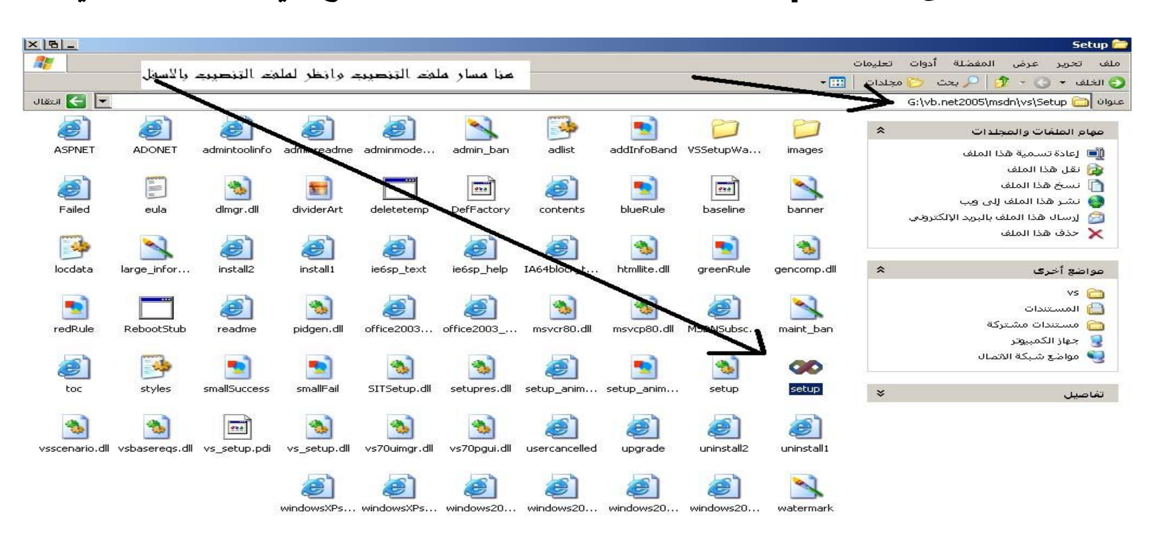
تنصيب (إعداد Visual Studio .Net 2005)

الصور التالية توضح عملية تنصيب Visual Studio .Net 2005

أولا نضغط على الملف Setup من مجلد التنصيب الموضح في الصورة التالي :

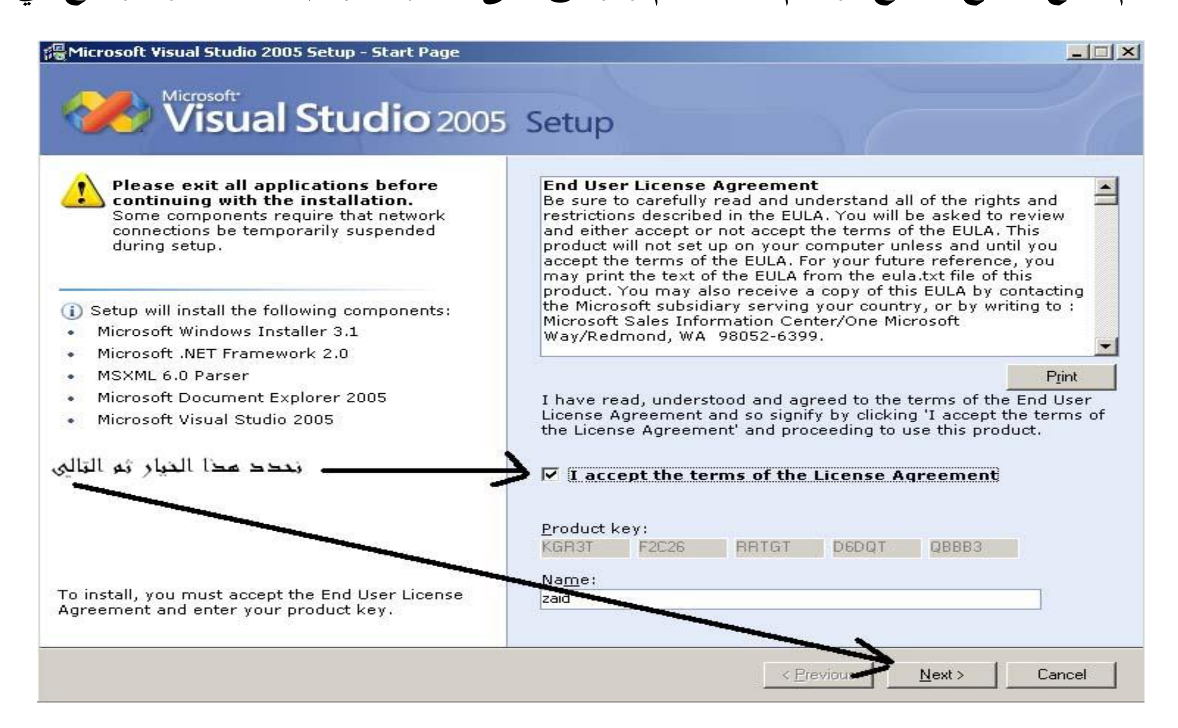




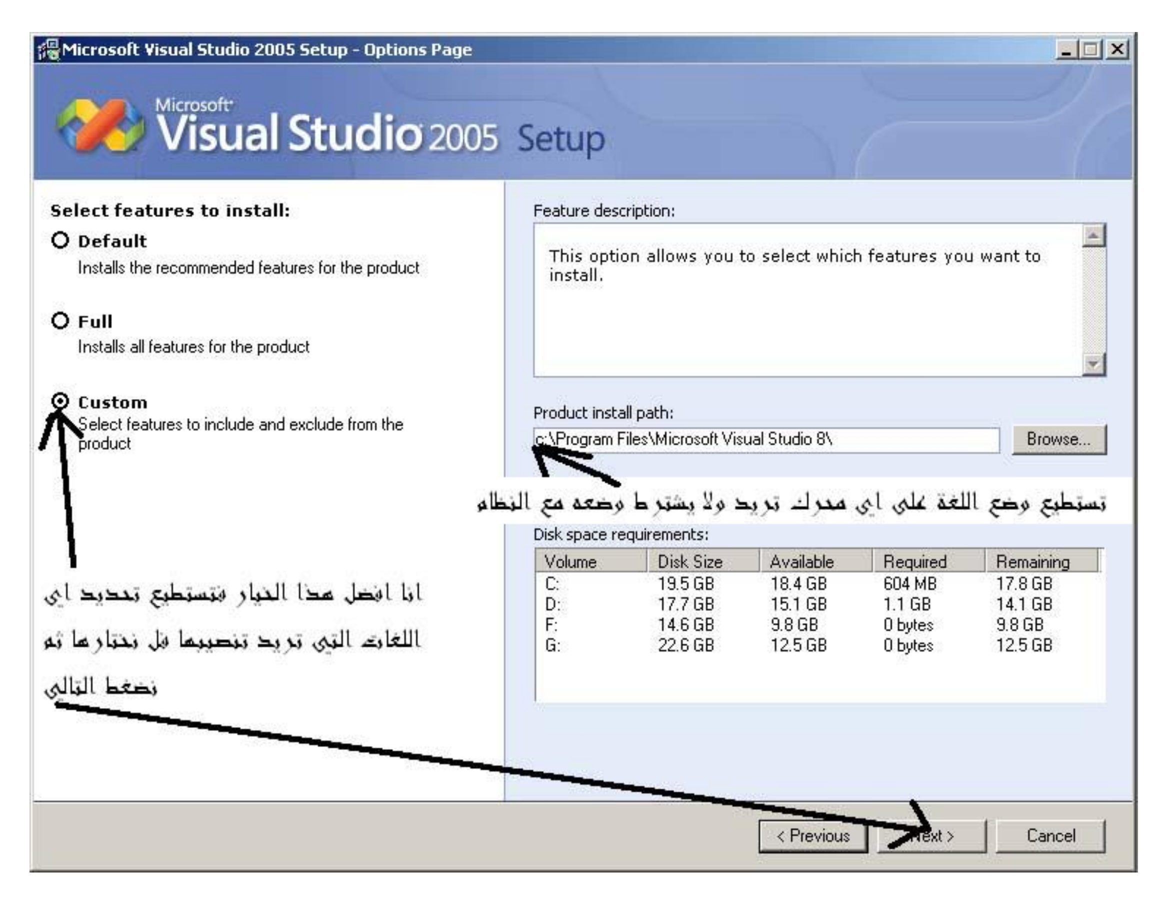
ثم تأتي الواجهة الترحيبية للتنصيب انتظر لحظات ثم اضغط Next



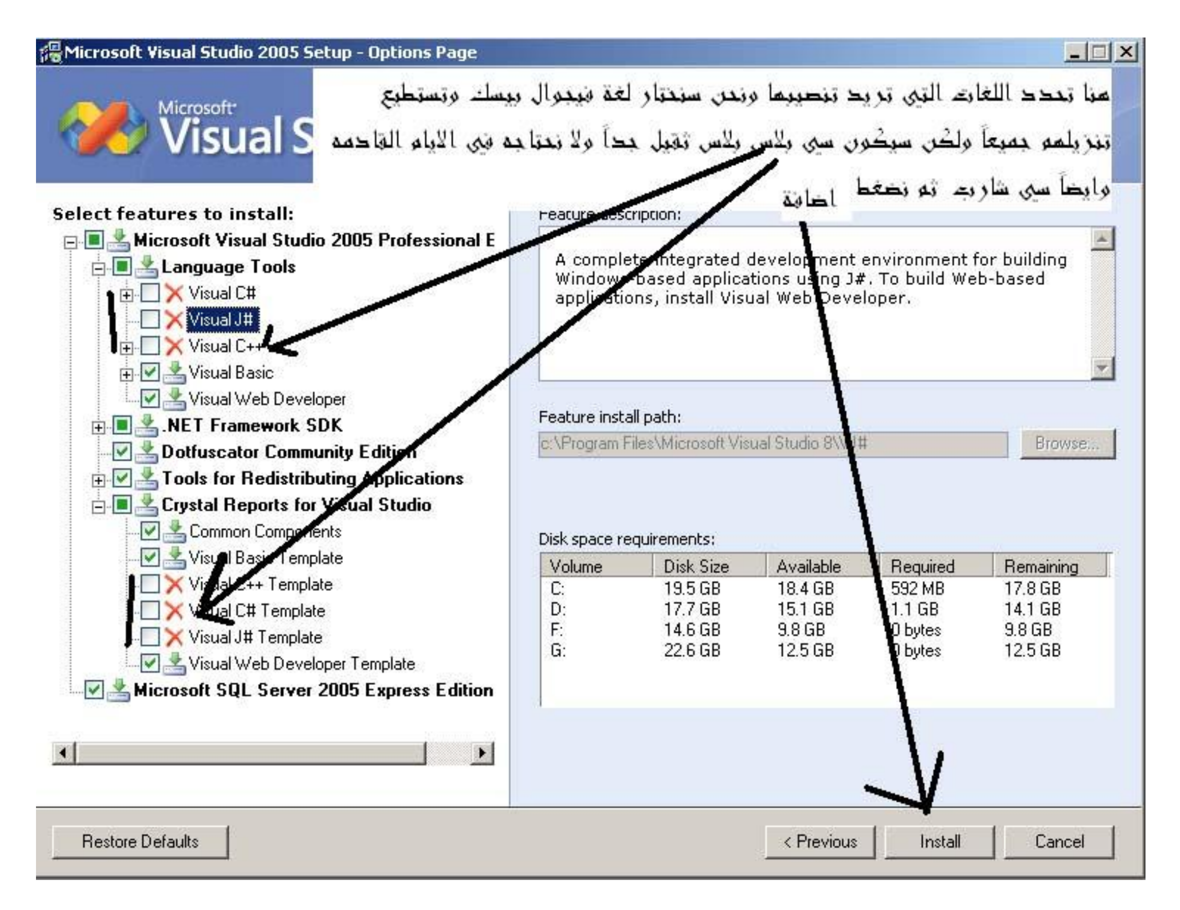
ثم نضع مفتاح المنتج واسم المستخدم ونوافق على اتفاقية الترخيص كما هو موضح في الصورة



ثم نختار طريقة التنصيب ونختار منها الخيار **Custome** كما هو موضح في الصورة التالية:



ومن هنا نختار اللغات المطلوبة ونزيل الاختيار عن باقي اللغات كما توضح الصورة



ومن ثم نكمل عملية التنصيب بشكل طبيعي وبإذن الله بدون مشاكل

بيئة تطوير Vb.net 2005

بيئة فيجوال بيسك دوت نت 2005أو كما يعرف IDEوهي اختصار للجملة Integrated development environment

VISUAL BASIC.NETهي احد اللغات المتاحة داخل برنامج VISUAL STUDIO.NE بالإضافة إلىVISUAL C#.NET

VISUAL J#.NET

VISUAL C++.NET

وبيئة VISUAL BASIC.NET هي بيئة متكاملة لإنشاء واختبار وتصحيح التطبيقات المتنوعة مثل :

WINDOWS APPLICATION

WEB APPLICATION

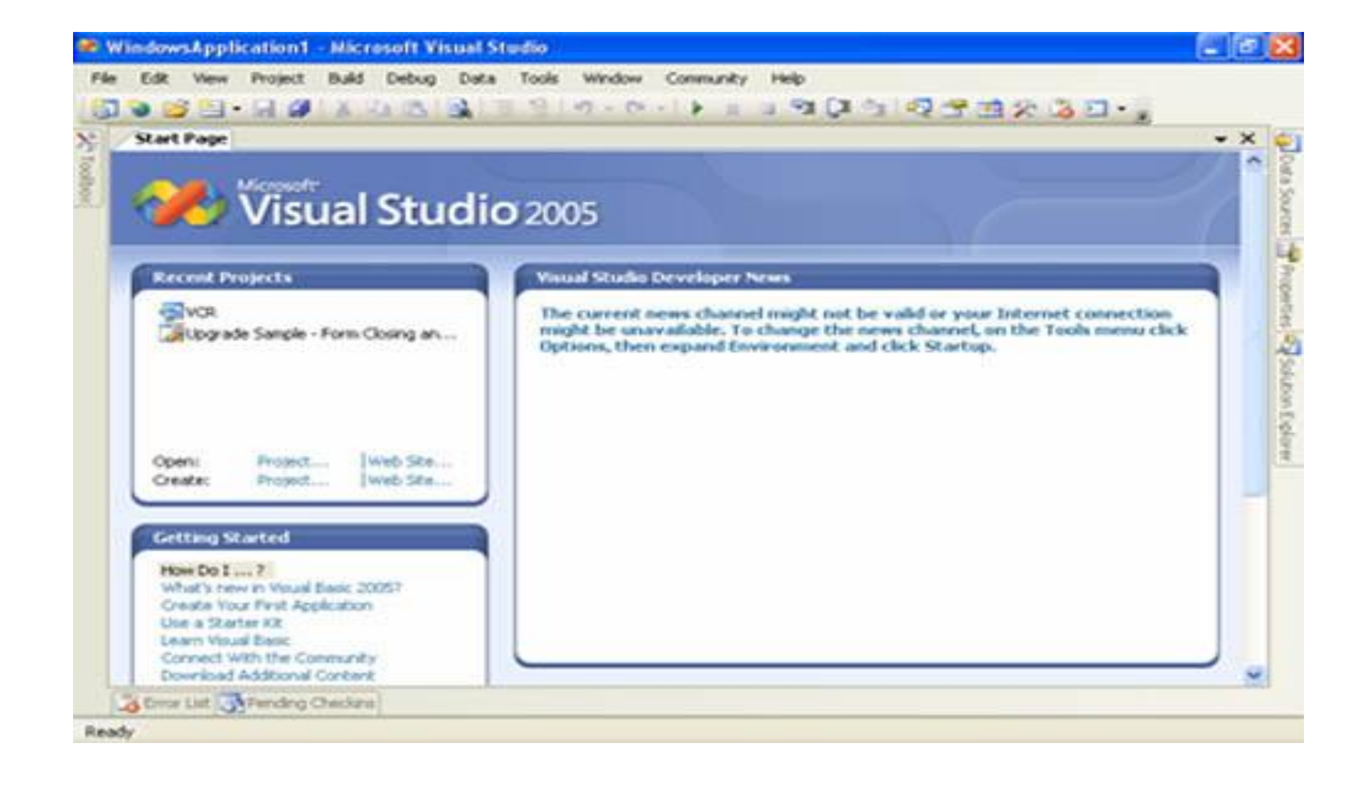
CLASSES AND CONTROL LIBRARY

CONSOLE APPLICATION

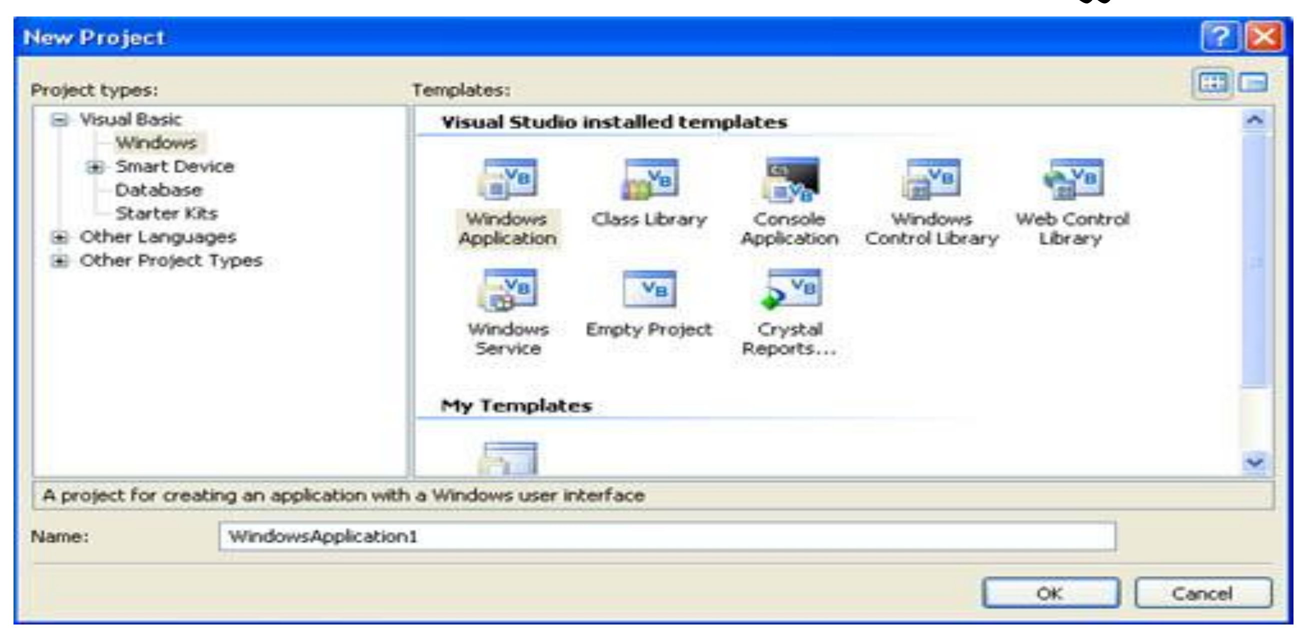
بالإضافة إلى غيرها من التطبيقات

مثل تطبيقات قواعد البيانات وتطبيقات الهاتف الكفي والهواتف الذكية..الخ

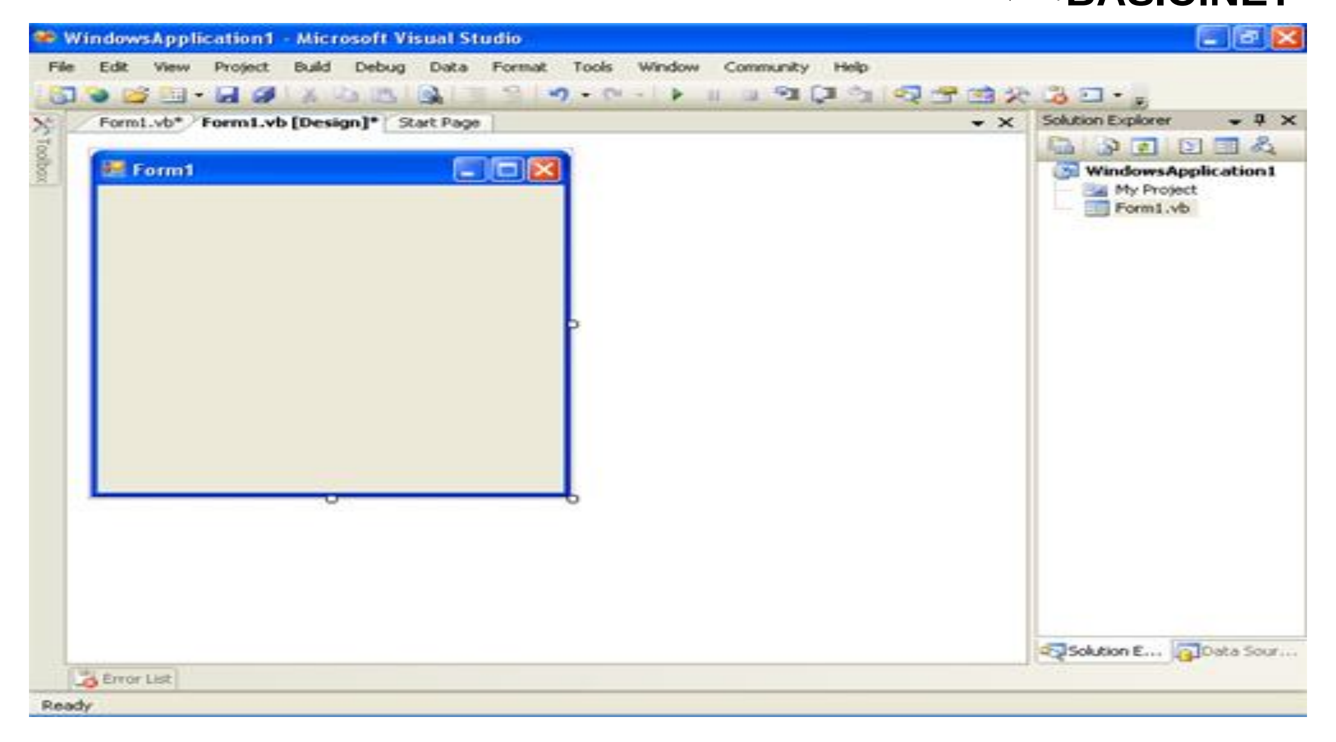
وعند فتح برنامج VISUAL STUDIO.NET للمرة الأولى سوف تظهر لك نافذة START PAGE



بها تظهر أخر تطبيقات تم إنشائها أو تعديلها ولكن هي لا تهمنا الآن قم بالضغط علىقائمة FILE و اختر منهاNew Project ستظهر لك نافذةNew Project لتختار منها التطبيق الذي تريد إنشائه لاحظ هذه الصورة



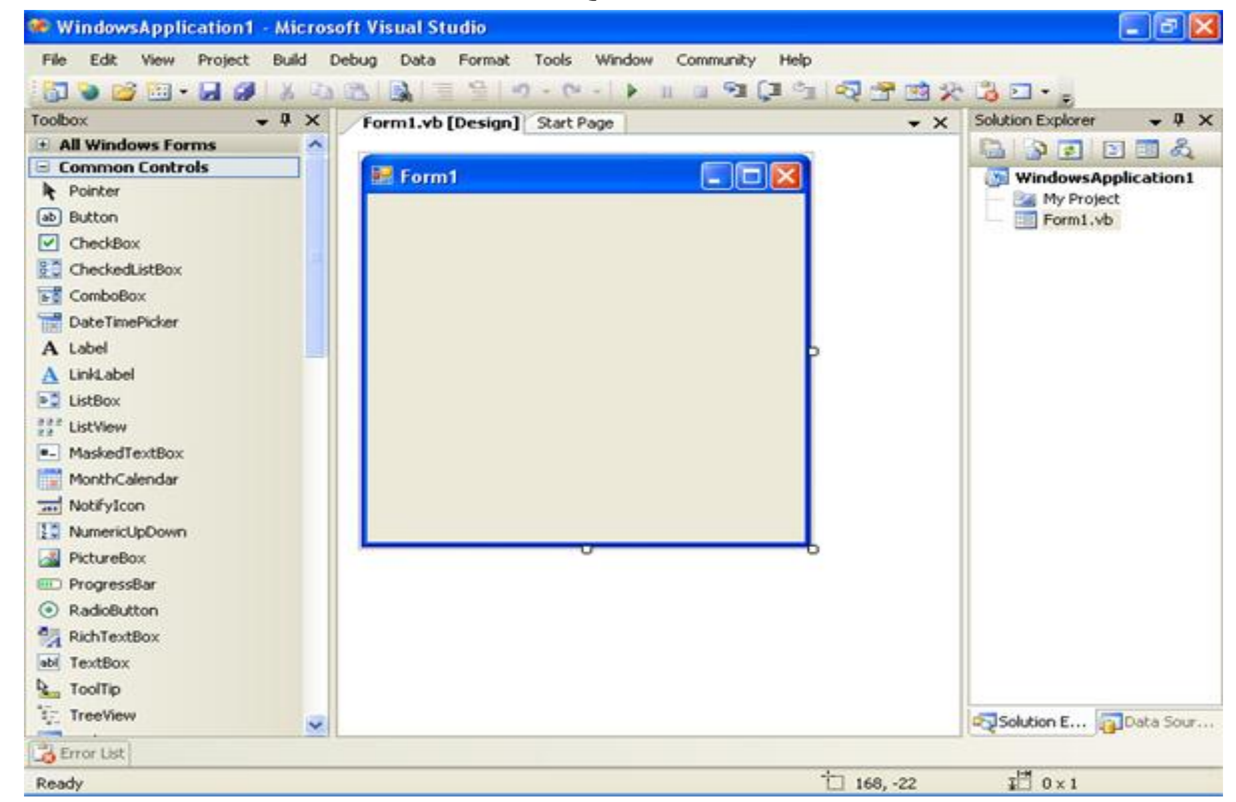
على اليسار يوجد نوع المشروع الذي نريد إنشاء احد تطبيقاته وعلى اليمين تظهر تطبيقات المشروع الذي حددناه وفي الأسفل النافذة اسم المشروع و يمكن أن نغيره إلى الاسم الذي نريده . نختار الآن احد تطبيقات VISUL BASIC.NETوسنختار من هذه النافذة WINDOWS APPLICATIONثم نضغط موافق سوف تظهر لنا قوائم وأدوات ونوافذ لغة VISUAL BASIC.NETهكذا



لإظهار نوافذ الخصائص أو الأدوات نقوم بوضع المؤشر على أسماء القوائم الموجودة جوانب برنامجVISUAL BASIC.NET ونلاحظ أننا إذا أبعدنا المؤشر فان القائمة تعاود الاختفاء فإذا أردنا تثبيتها ما علينا إلا الضغط على الزر الأوسط الموجود في أعلى القوائم الذي بهذا الشكل




وبعد تثبيت قائمة الأدوات سيظهر شكل البرنامج هكذا



والآن ماذا لاحظنا بالصورة نافذة المشروع الأساسية بالوسط ظاهرة بها فورم اسمه التلقائي Form1 وفي الجهة اليسرى تظهر قائمة الأدوات وسوف نتعرف على كل أداة موجودة بها في الدروس القادمة

ويظهر بالجهة اليمنى نافذة Solution Explorer ولكن ما هو الفرق بين Solution المشروع وال Project التطبيق ؟

Project أو التطبيق:

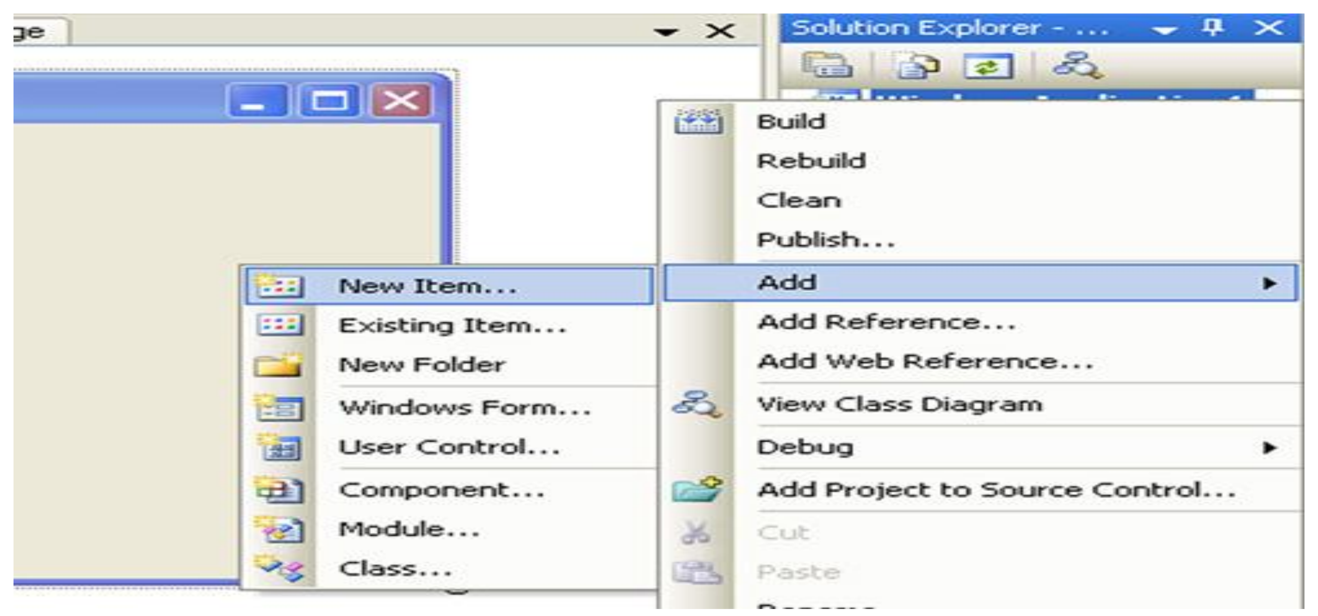
هو عبارة عن محتوى للمكونات الخاصة بتطبيق ما مثل النماذج Forms وكتل الاكواد البرمجية كال Modules أوClasses

ومع بداية ظهور لغة Vb.Net ظهر ما يسمي(Solution الحل) وفائدته تنظيمية لل Projects ويعتبر أل Solution مساحة العمل الخاصة بالدوت نت فمن الممكن أن يحتوي على أكثر من تطبيق Project بداخلة وتستطيع أيضا إضافة مكونات خارجة عن أطار الكود

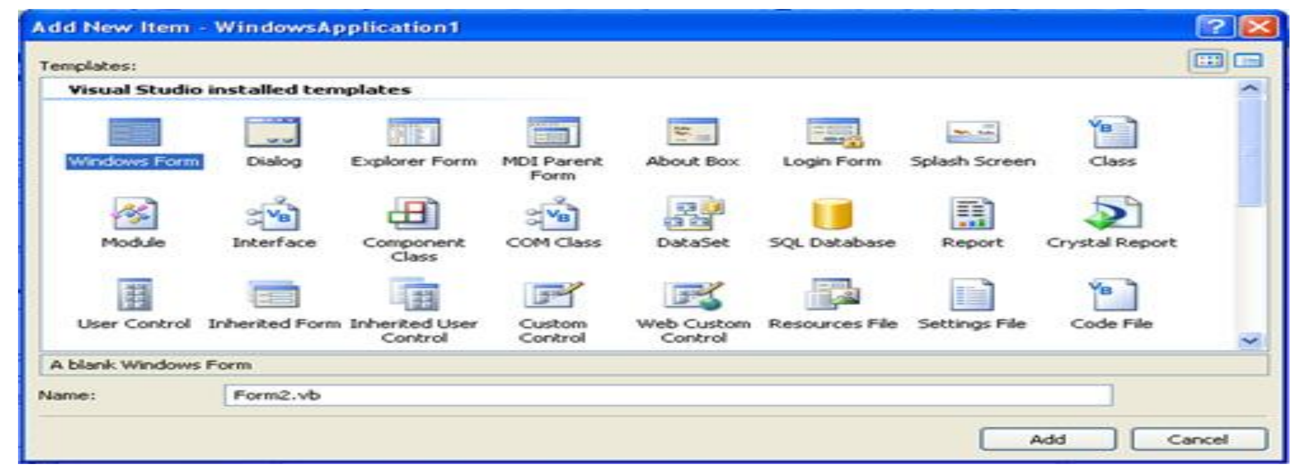
إلى Solution أي حتى ولو لم تتعلق بكود التطبيق كملف وورد أو صورة أو ملف صوتي وبالتالي تستطيع استخدامها داخل تطبيق الدوت نت

توجد النافذة Solution Explorer في الجهة اليمني ُكما يمكن إظهارها من قائمة View أو بالضغط على مفتاحي Ctrl+Alt+L وتعمل على عرض عناصر ومكونات التطبيق على هيئة قائمة شجرية في اعلي القائمة يظهر اسم التطبيق الحالي ويليه بقية مكونات التطبيق

ونستطيع إضافة عناصر إلى التطبيق الحالي من خلال النافذة Solution Explorer وذلك بالضغط على اسم المشروع بزر الماوس الأيمن ثم Add ومنة نختار New Item

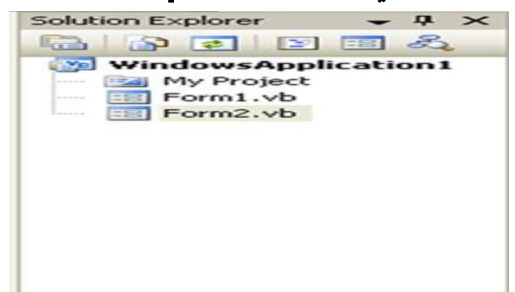


وبعد الضغط على New Item سوف تظهر لنا نافذة Add New Item



وبها العديد من تطبيقات VISUAL BASIC.NET نختار منها ما نريد اضافتة إلى تطبيقنا الحالي وهنا مثلا سوف نختار Windows Form قم بالتأشير علية ويمكننا تغيير اسم التطبيق قبل اضافتة من أسفل نافذة Add New Item أو نتركه كما هو ثم نضغط على الزر Add فنلاحظ إضافة Form أخرى إلى التطبيق تأخذ الاسم Form2

وتظهر في قائمة Solution Explorer كما بالصورة



الأدوات في الفيجوال بيسك دوت نت 2005

سيكون درسنا عن الأدوات خصائصها وإحداثها
الأدوات : هي عبارة عن عناصر برمجية لها مهام معينة تأخذ أشكال رسومية ولكل أداة عملها الخاص

*تصنف الأدوات إلى صنفين:

Control: وهو العنصر البرمجي الذي يأخذ شكل رسومي معين ويوضع على الفورم اثنا ء تصميم البرنامج ويقوم بتأدية مهمة معينة مثل Button , Textbox , Label
ولكل Control خصائص ووظائف تميزه عن غيره

Component: وهو شبيه بال Control فيما عدا:

1- لا يظهر في وقت التنفيذ أي وقت الاستعمال
2- يظهر اثنا تصميم البرنامج على شريط خاص به أسفل الفورم على شكل إيقونة
3- ولكن بالرغم من ذلك يمكن أن يظهر على الفورم إثناء التصميم مثل menu أو وقت التنفيذ مثل OpenFileDialog

*التعامل مع الأدوات:

الأدوات تساعد المبرمج إثناء تصميمه برنامج بقدر كبير فتقوم باختصار الكثير من الاكواد الصعبة والطويلة.ولكل أداة من أدوات الفيجوال بيسك دوت نت غرض معين ومهمة معينة وتقوم كل أداة بعمل مختلف عن الأداة الأخرى



والجدول التالي يوضح بعض الأدوات التي يمكن استخدامها في بيئة vb.net

اسم الأداة	شكل الأداة	الوظيفة
صندوق العنوان Label		تستخدم لعرض بيانات نصية على الشاشة (غالبا ما تستخدم كعنوان لأدوات أخرى) ولا يستطيع المستخدم أن يقوم بتعديل محتوى هذه الأداة
زر الأمـــــر Button		تستخدم لوضع الأوامر التنفيذية بداخلها وعند ضغط المستخدم على هذا الزر يتم تنفيذ الأمر أو الأوامر الموجودة بداخلها
صندوق النصوص Textbox		يستخدم لعرض واستقبال البيانات من المستخدم ويمكن للمستخدم تعديل النص الموجود بداخله (إذا سمح له بذلك)
صندوق الخيارات Checkbox		يستخدم عند وضع مجموعة من الخيارات ويمكن للمستخدم اختيار أكثر من خيار
زر اختيار Radiobutton		يستخدم عند وضع مجموعة من الخيارات ويمكن للمستخدم اختيار خيار واحد فقط(وعند اختيار أحد الأزرار تزول علامة الاختيار عن باقي الأزرار)
صندوق المجموعات Groupbox		يستخدم كحاوية لبعض الأدوات التي يتم التعامل معها كمجموعة واحدة (عندما نريد تحريك مجموعة من الأدوات معا توضع داخل صندوق المجموعات)
صندوق الصور Picturebox		يستخدم كوعاء توضع الصور بداخله
أداة القوائم Mainmenu		تقوم بعمل القوائم في البرنامج
أداة التاريخ والوقت DateTimePicker		تقوم بعرض التاريخ والوقت
المؤقت Timer		يقوم بتنفيذ أمر أو مجموعة أوامر كل فترة تحدد من قبل المبرمج
صندوق حوار خط FontDialog		صندوق تنسيق الخطوط
صندوق حوار ألوان ColorDialog		يستخدم في تنسيق الألوان

إنشـــــاء الأدوات:

***يمكنك إضافة أي أداة إلى برنامجك بإحدى هذه الطرق :**

1- النقر المزدوج على الأداة المراد إضافتها

2- لسحب والإفلات باستخدام الماوس

3- النقر نقرة واحدة على الأداة المراد إضافتها وثم النقر نقرة واحدة على الفورم وسيتم إضافتها

***ولكن لو أردنا إنشاء مجموعة كبيرة من الأدوات كعمل برنامج حاسبة به أزرار عديدة مثلا فإن الطرق السابقة لإضافة الأزرار ستكون متعبة قليلا .فما هو الحل باتري في رأيك ؟**

الحل هو الضغط المتواصل على مفتاح Ctrl ثم النقر نقرة واحدة باستخدام الماوس على الأداة التي تريد أضافه عددا منها ثم النقر على الفورم ستلاحظ في كل مرة تقوم بها بالنقر على الفورم إضافة الأداة مرة جديدة وباسم جديد وعندما تريد الانتهاء من إضافة الأداة فما عليك سوى إيقاف الضغط على مفتاح Ctrl والضغط على إيقونة مؤشر الماوس التي في اعلي الأدوات .

تجميع الأدوات داخل حاوية:

هنالك أدوات عملها الأساسي هو احتوى أدوات أخرى بداخلها مثل TabControl, Group Box, Panel,وباستخدامها تستطيع تجميع عدد من الأدوات في حاوية واحدة لغرض نقل وتحريك أو إخفاء وإظهار تلك الأدوات أو لأغراض أخرى. المهم في الأمر أننا باستخدام حاويات الأدوات نستطيع أن نتعامل مع مجموعة من الأدوات ككتلة واحدة .

هنالك عدة طرق لتضع الأدوات داخل حاوية container وذلك بسحب الأداة الموجود على الفورم إلى داخل container أو بتحديد ال container الذي قمت اضافتة مسبقا إلى الفورم ثم و من قائمة الأدوات تقوم بالنقر المزدوج على الأداة التي تريد أن تضيفها إلى container وستضاف بداخله مباشرة وليس على الفورم أو بواسطة السحب والإفلات من قائمة الأدوات إلى داخل إل. container أو بواسطة القص واللصق من على الفورم إلى داخل إل container .

خصائص الأدوات وكيفية التعامل معها

الخاصية : هي قيمة أو بعض القيم التابعة لأداة التحكم والتي تتحكم في عمل أو مظهر الأداة.

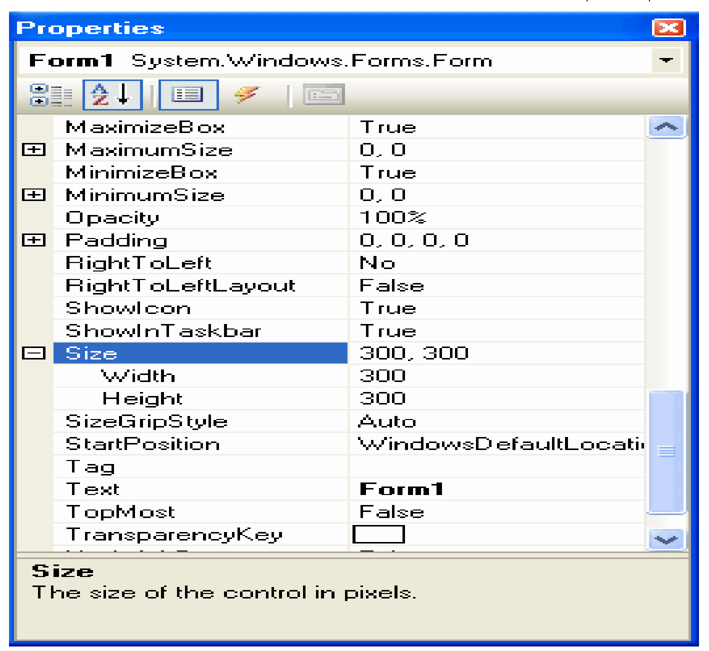
مثلا أداة التحكم Textbox تمتلك الخاصية Name وهي التي تتحكم في اسم الأداة البرمجي الذي يستخدمه المبرمج أثناء كتابة البرنامج وكذلك الخاصية Text وهذه الخاصية هي التي تتحكم في الكتابة التي ستظهر في Textbox وتوجد أيضا الخاصية Back Color وهي التي تتحكم في لون الخلفية وهكذا.

لكل أداة خصائص عديدة وتتشابه معظم الأدوات في الكثير من خصائصها ولكن هنالك خصائص لا توجد إلا في نوع محدد من الأدوات مثلا الأداة **Image List** تمتلك الخاصية**Images** التي تستطيع باستخدامها من اختيار العدد الذي تريد من الصور التي سوف تخزنها هذه الأداة لاستخدامها كإيقونات مثلا مع أدوات أخرى مثل استخدامها مع أداة العرض الشجري **Tree View .**

<u>الخصائص في وقت التصميم :</u>

لتعديل خصائص أي أداة من أدوات التحكم في وقت تصميم برنامجك يجب أولا أن تقوم بفتح نافذة الخصائص وتستطيع فتحها من القائمة **View** واختيار**Properties Windows** أو بالضغط على المفتاح **F4** من لوحة المفاتيح.

بعد فتح نافذة الخصائص يبقى أن نحدد الأداة التي نريد تعديل خصائصها وذلك يتم بطريقتين الأولى باختيار اسم الأداة من أعلى نافذة الخصائص أو بالنقر عليها نقرة واحدة باستخدام الماوس فتظهر لنا جميع خصائص تلك الأداة في نافذة الخصائص بعدها نقوم بالنقر بالماوس أمام اسم الخاصية المراد تغييرها وكتابة أو اختيار القيمة الجديدة لها.



<u>أنواع الخصائص:</u>

هنالك أنواع كثيرة للخصائص فهنالك الخصائص المركبة من أكثر من قيمة والخصائص المحصورة بين قيم محددة وثابتة كذلك هنالك الخصائص المجمعة كمجموعة وسوف نستعرض كل نوع من أنواع الخصائص على حدة كي نفهم كيفية التعامل معها.

1- الخصائص المركبة:

بعض من الخصائص تمتلك قيم مركبة مثلا الخاصية **Location** تتكون من الاحداثيان **X and Y**والخاصية **Size** تتكون من القيمتين **width and height** والخاصية **Font** مثلا تتكون من القيم **font's name, size,** وغيرها من خصائص الخطوط أي هنالك خصائص تتكون

من أكثر من قيمة ونلاحظ في الصورة السابقة أن هنالك خصائص أمامها علامة الزائد (+) تلك هي الخصائص المركبة . فلوا قمت بالنقر على هذه العلامة سوف تظهر لك قيم هذه الخاصية .

2- الخصائص المحصورة:

هي الخصائص التي تكون قيمها محصورة ومحددة وتكون قيمها على شكل قائمة سرد بها القيم الممكن إعطائها للخاصية ومثال على هذه الخصائص الخاصية Visible فهذه الخاصية تمتلك إحدى القيمتين True and False ولا يمكنك إعطائها قيم أخرى إلا إذا كانت القيم الأخرى مساوية للقيم الحالية مثلا القيمة True= -1 والقيمة. False= 0

3-الخصائص المجمعة:

هنالك خصائص تتكون قيمها من مجموعة من القيم أو العناصر مثلا أداة التحكم ListBox تمتلك الخاصية Items التي تتكون قيمتها من مجموعة من العناصر تلك التي تقوم أداة ListBoxبعرضها أو مثل الخاصية ImageList فهيا تتكون من مجموعة من الصور يتم عرضها بأدوات أخرى كذلك هنالك خصائص قيمها تتطلب وجود أداة أخرى فمثلا لعمل صور بأعلى عناصر الأداة TabControlيجب أن توجد الأداة ImageList وإعطاء الخاصية Imagesالتابعة لها مجموعة من الصور كي نقوم بعرضها كأيقونات في عناوين

والجدول التالي يوضح بعض أهم الخصائص

اسم الخاصية	الوظيفة
الاسم Name	تعبر عن اسم الأداة والذي يستخدم في الكود البرمجي
لون الخلفية Backcolor	تستخدم لتغيير لون خلفية الكائنات
لون الأمامية Foreclor	تستخدم لتغيير لون خط الكتابة
الخط Font	تستخدم في التحكم في الخط (نوعه – حجمه – نمطه)
صور الخلفية Backgroundimage	تستخدم لوضع صورة كخلفية للنموذج
شكل الإطار BorderStyle	تستخدم لتحديد شكل إطار الأداة
شكل المؤشر Cursor	تستخدم لتحديد شكل مؤشر الفأرة عند المرور على الأداة
التمكين Enabled	تستخدم لتمكين أو عدم تمكين المستخدم من التعامل مع الأداة
الظهور Visible	تستخدم لجعل الأداة مرئية أو غير مرئية للمستخدم
الحجم التلقائي Autosize	تستخدم لتغيير حجم الأداة تلقائيا حسب النص الموجود داخلها

تعدد الأسطر Multiline	تعدد الأسطر في أداة صندوق النصوص
التفاف النص Wordwrap	تجعل النص يلتف تلقائيا في أداة صندوق النصوص
محاذاة النص Textalign	محاذاة النص داخل الأداة
الكتابة (المحتوى) Text	تغيير النص الظاهر(محتوى) للأداة
حجم الأداة Size	التحكم في عرض وارتفاع الأداة
من اليمين لليسار RightToLeft	الكتابة من اليمين لليسار (مفيدة في التطبيقات العربية)

ويمكن تغيير خصائص الكائنات من خلال

1 -نافذة الخصائص (نشط الكائن المراد تغيير خصائصه بنقره ثم اضغط F4 ومن نافذة الخصائص اختار الخاصية واضبط القيمة المناسبة لها

2 -من خلال نافذة البرمجة : متبعا القاعدة التالية

ObjetName.propertyname= valuo

وسنطبق العديد من الأمثلة على أداة صندوق النصوص Textbox

لجعل صندوق النصوص متعدد الأسطر	Textbox1. Multiline = True
لإظهار أشرطة التمرير داخل صندوق النصوص	TextBox1.ScrollBars = ScrollBars.Vertical

بعض أوامر التعامل مع النموذج Form

كود إنـهاء البرنـامج:

كود:

```
End
```

كود أخفاء النـافذة:

كود:

```
Me.Hide()
```

كود بسيط تسمع من خلاله صوت ال Beep الخاص بالنظام:

كود:

```
Beep()
```

و الآن هذا الكود البسيط و الذي كنا نكتب بدل عنه صفحات ..و الآن أصبح خاصية من خصائص ال Form وهو الشفافية:

كود:

```
Me.Opacity = 0.5
```

طبعا بدل القيم لاحظ القيمة 0.5 أدخل قيم تتراوح بين 0 و 1 لتتحكم بدرجة شفافية النافذة .. إذا أدخلت 0 لن تظهر النافذة ستكون شفافة بالكامل ..و إذا أدخلت 1 لن تكون شفافة و إنما لن يحدث شيء لأنك تقريبا عمل ال Opacity 100% و على فكرة تستطيع أن تغير هذه الخاصية من خلال خصائص النافذة وقت التصميم و ليس التنفيذ أيضا..

بعض الخصائص الجديدة:

اذهب إلى خصائص النافذة Form ثم إلى الخاصية AutoScroll و اجعلها تساوي القيمة True ثم ضع بعض العناصر في النافذة بأماكن عديدة و شغل البرنامج...لاحظ كيف أنه عند تصغير أبعاد النافذة تظهر أشرطة تمرير أفقية و عمودية تستطيع من خلالها استعراض كامل أدوات النافذة..

إظهار و إخفاء النوافذ:

لقد اختلفت طريقة إظهار النوافذ في بيئة Net. حيث يجب أن نعرف عن نافذة جديدة ثم نقوم بإظهارها.. كالتالي:

كود:

```
Dim frm As New Form1
    frm.Show()
```

لاحظ كيف عرفنا بال Dim عن نافذة اسمها frm و أسندنا قيمة ال Form1 لها ..ثم أظهرنا النافذة بالأمر Show..

وأيضا تستطيع استخدام الأمر ShowDialog لإظهار النافذة كالتالي:

كود:

```
Dim frm As New Form1
```

```
frm.ShowDialog()
```

و الفرق بين ال Showو ال ShowDialogهو أنه في حالة ال ShowDialogتكون النافذة طاغية على بقية النوافذ أي أنها
الوحيدة المفعلة وهذه الطريقة محبذة دوما ..أما ال Showفتظهر النافذة دون أن تجعلها الأساسية ..لذلك إذا طلبت الأمر Showعدة مرات ستظهر لك عدة نسخ من النافذة نفسها..

<u>إغلاق النافذة</u>:-

كود:

```
Me.Close()
```

لاحظ أننا أغلقتاها و لم نخفيها ..لإخفاء النافذة يمكن كتابة الكود التالي :
كود:

```
Me.Hide()
```

طبعا ستسألني متى نستخدم هذا و متى نستخدم ذاك ..
الآن إذا كانت النافذة هي ليست النافذة الرئيسية التي ينطلق منها المشروع و هي عادة Form1فقم بإغلاق النافذة Close أما إذا كانت النافذة هي الرئيسية فقم بإخفائها لأنك لو أغلقتها ستعتبر اللغة أنك أغلقت البرنامج كله..

<u>عناصر لغة البيزك المرئى Vb.net2005</u>

تتكون اللغة من مجموعة من العناصر تمثل معا في نهاية الأمر جملة برمجية يفهمها الحاسب من خلال المترجم أو المفسر الخاص باللغة . وتتمثل تلك العناصر في :

1- الرموز الأساسية للغة : وفي لغتنا تتمثل تلك الرموز في (1) الحروف: وتتعرف اللغة على حروف اللغة الانجليزية من A إلى Z ولا يفرق مترجم اللغة بين حالة الأحرف سواء كانت حالة الحروف الكبير أو حالة الحروف الصغيرة (2) الأرقام : وتتعرف لغتنا على أرقام النظام العشري من 0 إلى 10 (3) العلامات الخاصة : وهي علامات ذات مدلول خاص بالنسبة للغة مثل * والتي تعني القيام بعملية الضرب .....الخ

2- التعبيرات : هناك العديد من التعبيرات ولكن سوف نهتم بـــــ :

أولا التعبيرات الحسابية : التعبير الحسابي هو تعبير يتكون من أرقام أو قيم عددية يفصل بينها مؤثر (معامل حسابي) ويكون الناتج قيمة عددية (رقم)

والجدول التالي يوضح أهم المعاملات الحسابية

المعامل	معناه	مثال	الناتج
^	أس	2^3	8
*	ضرب	2*3	6
/	قسمة	6/2	3
+	جمع	2+3	5
-	طرح	6-3	3
Mod	المتبقى من القسمة	11 Mod 3	2
\	القسمة الصحيحة	11\3	3

ويتم تنفيذ المعاملات السابقة(ترتيب تنفيذ المعاملات) وفقا للجدول التالي :

الترتيب	معناه	مثال	الناتج
1	الأقواس( )	(2+3)*7	35
2	الأس ^	2^3+1	9
3	الضرب والقسمة *, /, \, Mod	2+3 * 7	23
4	الجمع والطرح + , -	10-4 * 2+1	3

وإذا ظهرت عمليتين من نفس المستوى فإن البيزك يقوم بتنفيذها حسب ترتيب وجودها فى المعادلة الحسابية من اليسار إلى اليمين .

ثانيا التعبير العلاقي : هو تعبير يتكون من قيم عددية أو قيم حرفية أو كليهما ويفصل بين تلك القيم معامل علاقي ويكون ناتج التعبير العلاقي إما صواب True أو خطأ False

والجدول التالي يوضح المعاملات العلاقية :

المعامل	اسمه
=	يساوى
>	أكبر من
<	أصغر من
>=	أكبر من أو يساوى
<=	أصغر من أو يساوى

ويمكن استخدام المعاملات الشرطية للمقارنة بين أرقام أو سلاسل حروف ، وبالنسبة للحروف فهناك قواعد نلخصها في الآتي:

- الحروف الانجليزية الكبيرة Uppercase أقل في قيمتها من الحروف الإنجليزية الصغيرة Lowercase.
- تتم المقارنة بحسب الترتيب الأبجدي.
- الأرقام الموجودة في سلاسل الحروف أقل قيمة من الحروف نفسها.

هذا ويمكن الجمع بين أكثر من تعبير علاقي بواسطة استخدام أحد المعاملات المنطقية

المعاملات المنطقية logical Operators :

هي معاملات تتيح لنا الجمع بين اثنين أو أكثر من المعاملات الشرطية وهذه المعاملات تستخدم كلمات أساسية Keywords وليس رموزاً ويدعم البيزك المرئي العديد من المعاملات نذكر منها :

المعامل	الحالة
AND	يجب أن يكون كلا الجانبين صحيحا
OR	يجب أن يكون أحد الجانبين صحيحا أو كلا الجانبين صحيحا
NOT	ينقض القيمة الصحيحة

XOR	يجب أن يكون أحد الجانبين صحيحا ولكن ليس كلا الجانبين معا

ولكن ماذا لو اجتمعت كل هذه الأنواع من المعاملات معاً فى تعبير واحد ، فأى المعاملات تنفذ أولاً ؟؟

(1) ما بين الأقواس .

(2) أي عملية حسابية آسية إن وجدت .

(3) أي عملية حسابية بها تلك المعاملات (Mod,\,/,*)(ضرب أو قسمة).

(4) أي عملية حسابية بها معاملي (-,+)(جمع أو طرح).

(5) العملية الشرطية.

(6) العملية المنطقية NOT.

(7) العملية المنطقية AND.

(8) العملية المنطقية OR.

(9) العملية المنطقية XOR.

3- المتغيرات والثوابت

المتغير Variable :

هو موقع تخزيني في الذاكرة المؤقتة Ram يستخدم للاحتفاظ المؤقت بالبيانات التي قد تتغير أثناء تشغيل البرنامج

الثابت constant :

هو موقع تخزيني في الذاكرة المؤقتة Ram يستخدم للاحتفاظ المؤقت بالبيانات التي تظل ثابتة أثناء تشغيل البرنامج

أنواع المتغيرات والثوابت : يتم تصنيف المتغيرات حسب

أولا تصنيف المتغيرات حسب نوع البيانات :

البيانات Data : تنقسم إلى بيانات رقمية وبيانات نصية ومعيار التفرقة بين النوعين هو مدى خضوع البيان للعمليات الحسابية (جمع - طرح – ضرب - ....) فالبيانات الرقمية تخضع لكافة العمليات الحسابية أما البيانات النصية فلا تخضع للعمليات الحسابية .

ويتم تقسيم المتغيرات وفقا لنوع البيان الذي بداخله إلى:

Data type	Size	Range	Sample usage
Short	16-bit	-32,768 through 32,767	Dim Birds As Short Birds = 12500
Integer	32-bit	-2,147,483,648 through 2,147,483,647	Dim Insects As Integer Insects = 37500000
Long	64-bit	-9,223,372,036,854,775,808 to 9,223,372,036,854,775,807	Dim WorldPop As Long WorldPop = 4800000004
Single	32-bit floating point	-3.4028235E38 through 3.4028235E38	Dim Price As Single Price = 899.99
Double	64-bit floating point	-1.79769313486231E308 through 1.79769313486231E308	Dim Pi As Double Pi = 3.1415926535
Decimal	128-bit	values up to +/-79,228 x 1024	Dim Debt As Decimal Debt = 7600300.50
Byte	8-bit	0 through 255 (no negative numbers)	Dim RetKey As Byte RetKey = 13
Char	16-bit	Any Unicode symbol in the range 0–65,535	Dim UnicodeChar As Char UnicodeChar = "Ä"
String	Usually 16-bits per	0 to approximately 2 billion 16-bit Unicode characters	Dim Dog As String Dog = "pointer"

Data type	Size	Range	Sample usage
	character		
Boolean	16-bit	True or False (during conversions, 0 is converted to False, other values to True)	Dim Flag as Boolean Flag = True
Date	64-bit	January 1, 0001, through December 31, 9999	Dim Birthday as Date Birthday = #3/1/1963#
Object	32-bit	Any type can be stored in a variable of type Object	Dim MyApp As Object MyApp = CreateObject _ ("Word.Application")

ثانياً تصنيف المتغيرات حسب مدى المتغير :

يقصد بمدى المتغير الفترة الزمنية التي يظل المتغير معروفا فيها ومحتفظا بقيمته وعلى ذلك الأساس يمكن تقسيم المتغيرات إلى :

1. **متغير إجراء (متغير يكون معروفا فقط في الإجراء الذي تم الإعلان عنه فيه ويكون الإعلان بالكلمة المحجوزة Dim**
2. **متغير عام على مستوى النموذج (متغير يكون معروفا فقط في النموذج الذي تم الإعلان عنه فيه ويكون الإعلان بالكلمة المحجوزة Private)**
3. **متغير عام على مستوى المشروع (متغير يكون معروفا فقط في المشروع الذي تم الإعلان عنه فيه ويكون الإعلان بالكلمة المحجوزة friend)**

الفرق بين أنواع التعريفات:
Dim:لو وضعته في كود داخل عنصر سيقتصر مدى التعريف على الكود فقط
Private :يكون مجال التعريف على كامل أل Formو لاحظ أنه لا يصل مجاله إلى نافذة أخرى ..و لا يعمل تحت كود عنصر و إنما فقط في أل.Form Class
Friend: مجال تعريفه يتعدى أكثر من...Form و لا يعمل تحت كود عنصر و إنما فقط في ال.Form Class

Public: يشمل مجاله كل المشاريع الموجودة في Solution Explorer أي هو الأقوى و الأعم..

والآن كيف يتم الإعلان عن المتغيرات عن متغير:

سأقوم الآن بالتعريف عن متغير رقمي :

كود:

```
Dim a As Integer
```

لاحظوا ال Dimتخبر اللغة أن هناك شيء سيتم تعريفه و بعدها a و هي اسم المتغير تستطيع أن تكتب اسمك مثلا (هناك قواعد لكتابة المتغيرات مثل أن لابد تبدأ بحرف وأن لاتحتوي فراغات .....) و بعدها لو وقفت و لم تكتب شيء ستعتبر اللغة أنك عرفت عن متغير من نوع Object عام يمكن أن يحمل أي شيء في ما بعد و هذا ما يحبه المبتدأ ولكن لحظة لكل شيء ثمنه فحجم هذاالObjectفي الذاكرة أكبر بكثير من النوع المحدد مثل Integer أو..String
و الآن بعد a كتبنا الكلمة المحجوزة as يعني (هذا المتغير من نوع..) و كلمة ASأصلا تعني كـ أو مثل ..و أخيرا نوع المتغير وهو متغير رقمي من نوع) Integer لن أدخل في تفاصيل الفرق بينها و بين Int32 و غيرها)..و أساس بعد أن تكتب كلمةAs تظهر لك قائمة أنواع تختار منها ..

مثلا:

: Stringنص
: Fontخط
: Bitmapصورة
: Colorلون
: Booleanيحمل قيمتين إما True أو..False

و الآن بعد أن عرفنا كيف نعرف المتغير لنعرف الثابت.

كود:

```
Const a As Integer = 10
```

لاحظوا الذي اختلف هو إسناد القيمة يعني بما أنه رقم كتبنا قيمة 10 ولو كان String لكنا كتبنا مثلا "vb" يعني أي نص بين علامتي تنصيص ولو كان Boolean لكتبنا إما True أو Falseو هكذا.الآن لنقل أننا نريد أن نستعمل هذا المتغير أو الثابت ..لاحظوا سأقوم بنقل محتوى المتغير إلى عنوان النافذة.

كود:

```
Const a As String = "first"
     Me. Text = a
```

طبعا لو أردنا استخدام متغير سنقوم بكتابة قيمة له مثلا.
كود:

```
Dim a As String
     a = "vb"
     Me.Text = a
```

طبعا لو أردنا أن نعرف أكثر من متغير من نفس النوع
كود:

```
Dim a ,b ,c As String
```

طبعا لو أردنا أن نعرف أكثر من متغير من أنواع مختلفة
كود:

```
Dim a As String , b as integer ,c as byte
```

5- جمل البيزك المرئي :

من خلال السطور التالية سوف ندرس جملتين من جمل اللغة وهما جملة الشرط (التفريع) وجملة التكرار (الدوارة) وهما من الجمل التي تستخدم للتحكم في سير البرنامج (هناك عدة أوامر تحدد خط سير البرنامج فيمكن أن يكون السير في البرنامج على التوالي أي سطر يتبع سطر, ويمكن أن يقفز البيزك المرئي إلى سطر في آخر البرنامج اعتمادا على جملة شرطية أو يقوم بعمل نفس العملية عدة مرات, تلك الإجراءات تحدد بواسطة الأوامر التي تتحكم في سير البرنامج ، فما هي تلك الأوامر؟)

1- أداة الشرط IF .. THEN:

تعتمد فكرة جملة الشرط على اختبار شرط معين (أو أكثر) في حالة تحقق الشرط (إذا كان الشرط صحيح) يتم تنفيذ أمر (أو مجموعة من ألأوامر) وفي حالة عدم تحقق الشرط (الشرط خطأ) يتم تنفيذ أمر (أو مجموعة أوامر أخرى)

وجملة If .....Then .....Else من أشهر الأوامر في البيزك المرئي ولها عدة أشكال :

الشكل الأول

IF < شرط > Then

الشكل الثاني

IF < شرط > Then

<سطر أو مجموعة سطور أوامر>

End IF

والشرط هنا هو تعبير يعود بقيمة صحيحة أو خطأ أو يكون تعبيراً طويلاً يتضمن عدة معاملات منطقية.

لاحظ من الشكل الأول والثاني لجملة الشرط If أن Else وما يليها اختياري أي يمكن كتابته أو عدم كتابته حسب الحالة التي نحن بصددها

الشكل الثالث

If < شرط > Then أمــــر Else أمر

لاحظ أن الأمر أو مجموعة الأوامر بعد Then تنفذ فقط في حالة تحقق الشرط وأن الأمر او مجموعة الأوامر بعد Else تنفذ فقط في حالة عدم تحقق الشرط

الشكل الرابع

(2). IF < شرط > Then

<سطر أو مجموعة سطور أوامر>

Else

<سطر أو مجموعة سطور أوامر>

End IF

الصيغة IF ... Then ...Else تحتوى على مجموعتين من الأوامر بحيث يتم تنفيذ المجموعة الأولى من الأوامر إذا تحقق الشرط ويتم تنفيذ المجموعة الثانية إذا لم يتحقق الشرط.

التداخل في صيغة IF .. THEN:

يمكن أن وضع تركيب IF داخل تركيب IF أخرى وهو ما يسمى بالتداخل Nesting.

IF < شرط1 > Then

IF < شرط2 > Then

<سطر أو مجموعة سطور أوامر>

End IF

<سطر أو مجموعة سطور أوامر>

End IF

التداخل في صيغة IF .. THEN ... ELSE:

إذا أردنا استخدام تركيب IF ... Then ...Else داخل تركيب IF ... Then ...Else آخر ,كما في المثال التالي :-

IF < شرط1 > Then

<سطر أو مجموعة سطور أوامر>

Elseif < شرط2 > Then

<سطر أو مجموعة سطور أوامر>

Else

<سطر أو مجموعة سطور أوامر>

End IF

وفى هذه الحالة تعتبر سطور الأوامر بدائل لبعضها ، أي لا تنفذ إلا مجموعة واحدة فقط.

2- تركيب SELECT ... CASE:

تعتبر Select .. Case مناسبة أكثر عند اختيار عدة شروط في آن واحد وبدلا من استخدام عدة تركيبات متداخلة من If .. then .. Else والذي سيؤدى إلى صعوبة فهم البرنامج ,لذا عند وجود عدة اختيارات في جملة نستخدم Select .. Case صيغتها كالأتي:

```
Select Case     < تعبير >
        Case 1           <قيمة>
        <سطر أو مجموعة سطور أوامر>
        Case 2           <قيمة>
        <سطر أو مجموعة سطور أوامر>
        Case else
        <سطر أو مجموعة سطور أوامر>
End Select
```

أى يتم وضع التعبير المراد اختياره في بداية الصيغة ثم وضع عدة قيم وراء كل قيمة مجموعة الأوامر, وسوف يقوم البرنامج بتنفيذ مجموعة الأوامر التي تلي هذه القيمة ، وإن لم يجد البرنامج أي قيمة متوافقة فسوف ينفذ الأوامر في عبارة Case else وتلك العبارة اختيارية ، فإن لم تكن موجودة فإن البرنامج لن ينفذ أي أمر وسوف يخرج من الصيغة بأكملها .

ويمكن استخدام شكل آخر

```
Select Case     < تعبير >
        Case Is     < تعبير >
        <سطر أو مجموعة سطور أوامر>
End Select
```

ويمكن استخدام شكل ثالث لتحديد نطاق من القيم

```
Select Case     < تعبير >
        Case 1          < قيمة1 >   TO    < قيمة2 >
        <سطر أو مجموعة سطور أوامر>
        Case  else
        <سطر أو مجموعة سطور أوامر>



End Select
```

وكذلك يمكن جمع الصيغ المختلفة لـ Select Case

```
Select case x
Case 101
Case 102 to 105
Case is > 105
End select
```

ملحوظة : هناك بعض الحالات لا يمكن فيها استخدام تركيب Select .. Case مثل استخدام Select .. Case المعاملات المنطقية And, Or, Xor ، فى هذه الحالة يجب استخدام IF .. THEN .. ELSE.

3- التكرار (Looping)

يتيح البيزك المرئي تكرار مجموعة من الأوامر باستخدام ما يسمى الـ Loops ، وهو تكرار مجموعة من الأوامر حسب عدد المرات المحددة من مصمم البرنامج أو عند تحقق شرط معين.وهناك عدة أوامر تحدد التكرار .

<u>أ- التكرار باستخدام Do</u>

وهو أبسط أنواع التكرارات وله عدة صور كما يلي:

الشكل الأول

DO WHILE <شرط>

<سطر أو مجموعة سطور أوامر>

Loop

الشكل الثاني

DO

<سطر أو مجموعة سطور أوامر>

LOOP WHILE <شرط>

الشكل الثالث

DO UNTIL <شرط>

<سطر أو مجموعة سطور أوامر>

LOOP

الشكل الرابع

DO

<سطر أو مجموعة سطور أوامر>

LOOP UNTIL <شرط>

إن كلمة الشرط التي تظهر داخل تكرار DO هي أي تعبير أو أداة أو قيمة منطقية يمكن تقييمها إلى قيمة صحيح True أو خطأ False.

- **موقع الشرط**

إذا ظهر الشرط في بداية التكرار فإن البرنامج يقيم الشرط أولاً قبل تنفيذ أي أمر موجود داخل التكرار، فإن وجد أن الشرط غير متحقق يخرج من التكرار دون تنفيذه .

أما إذا ظهر الشرط في نهاية التكرار فإن البرنامج ينفذ الأوامر الموجودة داخل التكرار مرة واحدة على الأقل ثم يقيم الشرط فإن وجد أن الشرط متحقق يكرر تنفيذ الأوامر ، وإن وجد أن الشرط غير متحقق يخرج من التكرار.

- **طبيعة الشرط**

يمكن جعل تكرار DO يستمر في حالتين هما :

في الحالة الأولى يستمر التكرار ما دام الشرط صحيحاً ويتوقف عندما يصبح الشرط خاطئاً .

في الحالة الثانية يستمر التكرار ما دام الشرط خاطئا ويتوقف عندما يصبح الشرط صحيحاً .

يلاحظ أن :

- لابد من وجود أمر بالبرنامج لتعديل قيمة الشرط حتى يتوقف التكرارفي لحظة محددة.
- يستحسن استخدام الصيغة التي بها الشرط فى نهاية التكرار حيث تقبل معلومة من المستخدم, حتى يتم تنفيذ التكرار مرة واحدة على الأقل ثم اختبار المعلومات التي أدخلها المستخدم.

<u>ب- التكرار باستخدام FOR</u>

يدعم البيزك المرئي نوعاً آخر من التكرارات يسمى FOR يتم تنفيذه عددا محددا من المرات ، ويتخذ الشكل التالي :

FOR <متغير التكرار> = البداية **TO** النهاية **STEP** < الزيادة أو النقص>

<سطر أو مجموعة سطور أوامر>

Next

متغير التكرار :

هو متغير رقمي يتحكم في تنفيذ التكرار ويمكن تسميته بأي اسم,وعند بداية البرنامج يتم تهيئة متغير التكرار حسب رقم **البداية** الذي يحدد في رأس التكرار وغالباً يبدأ من (1) في كل مرة يتم تنفيذ مجموعة الأوامر يقوم البيزك المرئي بزيادة قيمة المتغير بمقدار الزيـــادة التي تحددها بعد كلمة (STEP) ، وكلمة STEP اختيارية فإذا لم تحدد فإن البيزك سيقوم بزيادة المتغير بمقدار (1), عندما يصبح المتغير أكبر من رقم النهاية فإن البرنامج يتوقف عن تكرار الأوامر ويبدأ في تنفيذ الأوامر التي تلي كلمة Next ,كلمة NEXT هي التي تحدد نهاية التكرار وذلك إذا وصل البرنامج لكلمة Next وكانت قيمة متغير التكرار أقل من رقم النهاية يتم زيادة قيمة المتغير بقيمة الزيادة يتكرر تنفيذ الأوامر مرة أخرى.

التداخل فى تكرارات FOR

يتيح البيزك المرئي تداخل اثنين أو أكثر من تكرارات For ، تكون الصيغة كالتالي:

```
FOR X = 1 TO 4
    FOR Y = 1 TO 10
    <سطر أو مجموعة سطور أوامر>
    NEXT Y
<سطر أو مجموعة سطور أوامر>
NEXT X
```

وفى هذا المثال يتم تنفيذ مجموعة الأوامر 40 مرة لأن التكرار الخارجي يتم تنفيذه 4 مرات ، والتكرار الداخلي يتم تنفيذه 10 مرات.لتحديد نهاية التكرار في الجمل المكونة من تركيب تكرار FOR لابد من كتابة اسم متغير التكرار بعد كلمة NEXT.

البرامج

البرنامج الأول

الهدف : التعامل مع نوافذ بيئة vb .net

عمل البرنامج : 1- يطلب من مستخدم البرنامج إدخال اسم

2- عند نقر زر أمر (ترحيب) تظهر رسالة ترحيب بالاسم الذي أدخله المستخدم

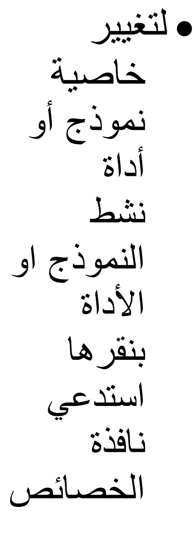
3- عند نقر زر أمر (مسح النص) يتم مسح الاسم الذي أدخله المستخدم ورسالة الترحيب

4- عند نقر زر أمر (إنهاء البرنامج) يتم إنهاء العمل بالبرنامج

الأدوات : عدد 3 (ثلاثة) أزرار أوامر - صندوق نصوص – عدد (2) صندوقي عنوان

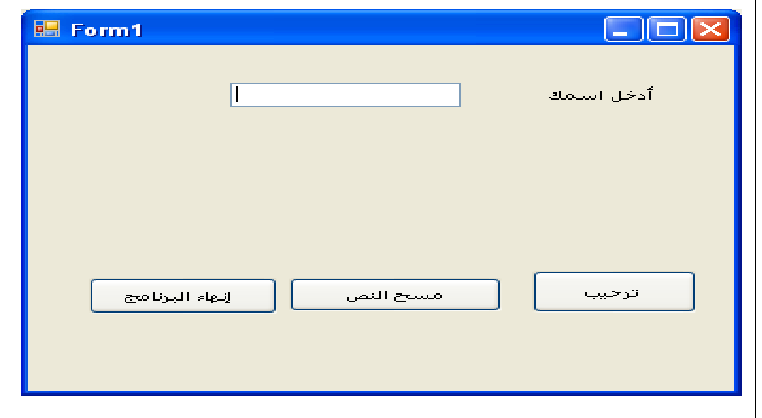
خطوات عمل البرنامج

<table>
<tr><th>الخطوة</th><th>الشرح التفصيلي</th></tr>
<tr><td>1 -تصميم واجهة المستخدم (وضع الأدوات على النموذج)</td><td></td></tr>
<tr><td>2 -ضبط خصائص الكائنات (تغيير بعض خصائص الأدوات)</td><td>
<table>
<tr><th>اسم الأداة</th><th>الخاصية</th><th>القيمة</th></tr>
<tr><td>Button1</td><td>Text</td><td>ترحيب</td></tr>
</table>
</td></tr>
</table>



• لتغيير خاصية نموذج أو أداة نشط النموذج او الأداة بنقرها استدعي نافذة الخصائص

مسح النص	Text	Button2
أدخل اسمك	Text	Label1
	Text	Label2
Yes	RightToLeft	Form1



3 كتابة تعليمات البرنامج

1 كتابة تعليمات زر الترحيب

```
Private Sub
 Label2.Text= "مرحبا بك في عالم البيزك المرئي يا " & TextBox1.Text
 End Sub
```

2 كتابة تعليمات زر مسح النص

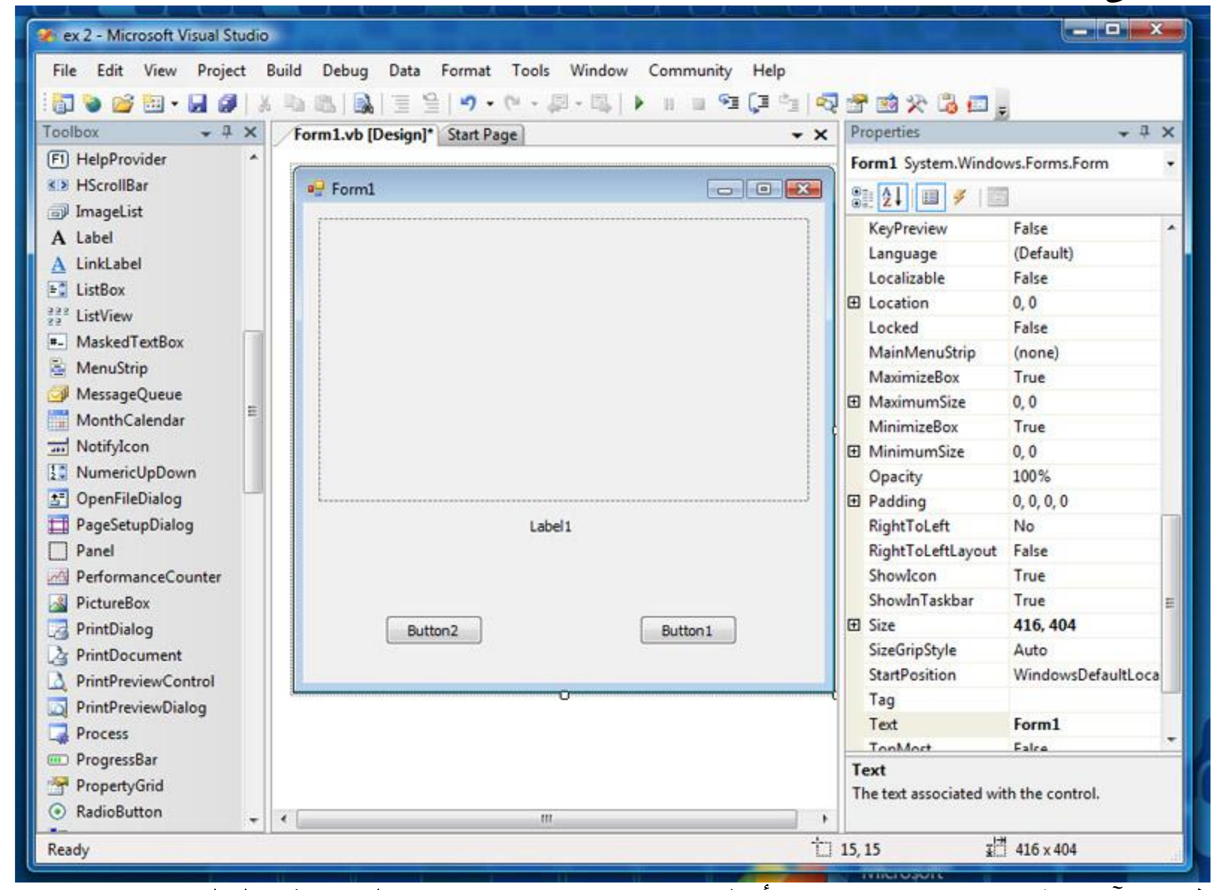
```
Private Sub
        TextBox1.Clear()
        Label2.Text = ""
        Textbox1.foucs()
End Sub
```

	3 ـ كتابة تعليمات زر إنهاءالبرنامج Private Sub End End Sub

البرنامج الثانى :إدراج صورة وعرضها داخل الفورم

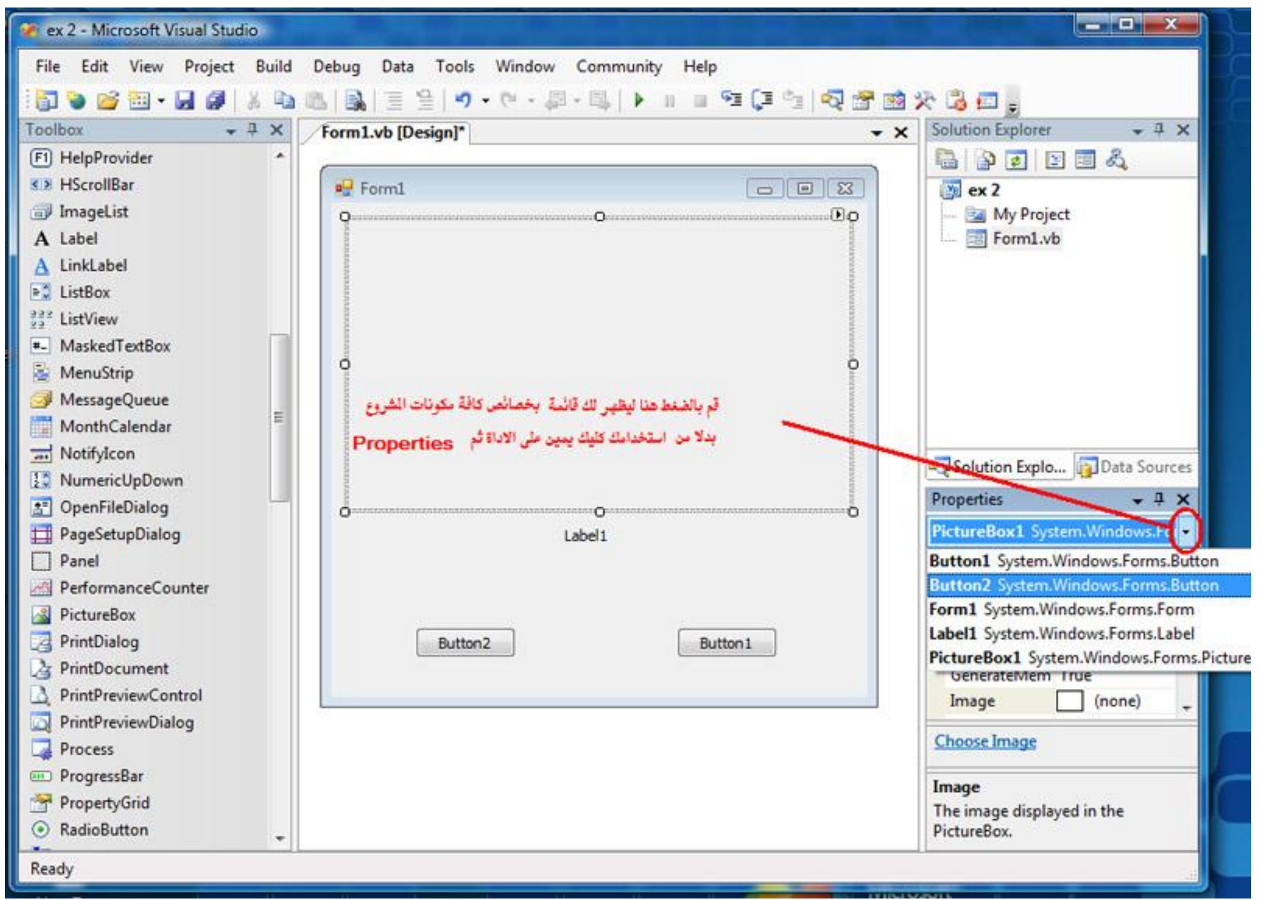
1- نقوم بعمل مشروع جديد

2- نقوم بإدراج عدد 2 Botton واحد Label و أداة PictureBox داخل الفورم ونقوم بتنظيمها فى شكل مقبول

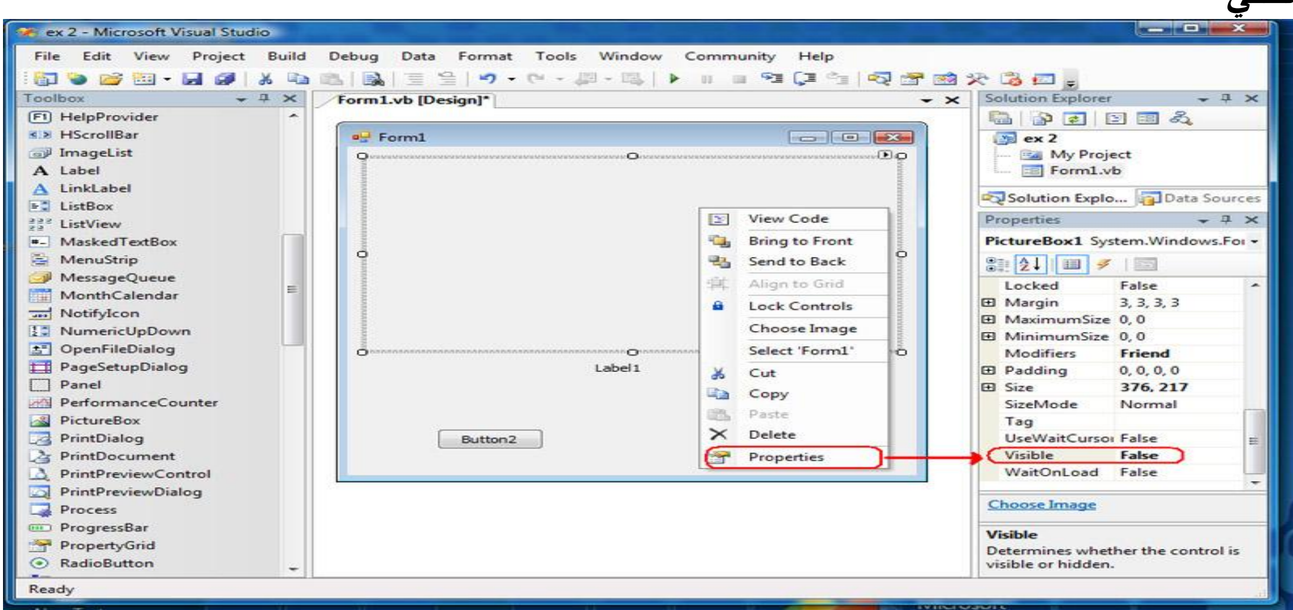


3- نقوم الآن بتغيير بعض خصائص أداة PictureBox مربع الصورة كالتالي

-كليك يمين بالماوس على مربع الصورة PictureBox ثم Properties أو نقوم بالوصول إليها عن طريق قائمة Properties في أقصى اليمين كما هو موضح بالشكل التالي



-قم بتغيير Visible لتصبح False بدلا من True ومعنى visible يعنى مرئي أو يمكن مشاهدته ومعنى true يعني حقيقي ووجود هذه الخاصية في الوضع true يجعلها مرئية باستمرار أما الوضع false الذي يعني كاذب أو مزيف فيعنى أن ما بداخل أداة الصورة سيكون مخفي

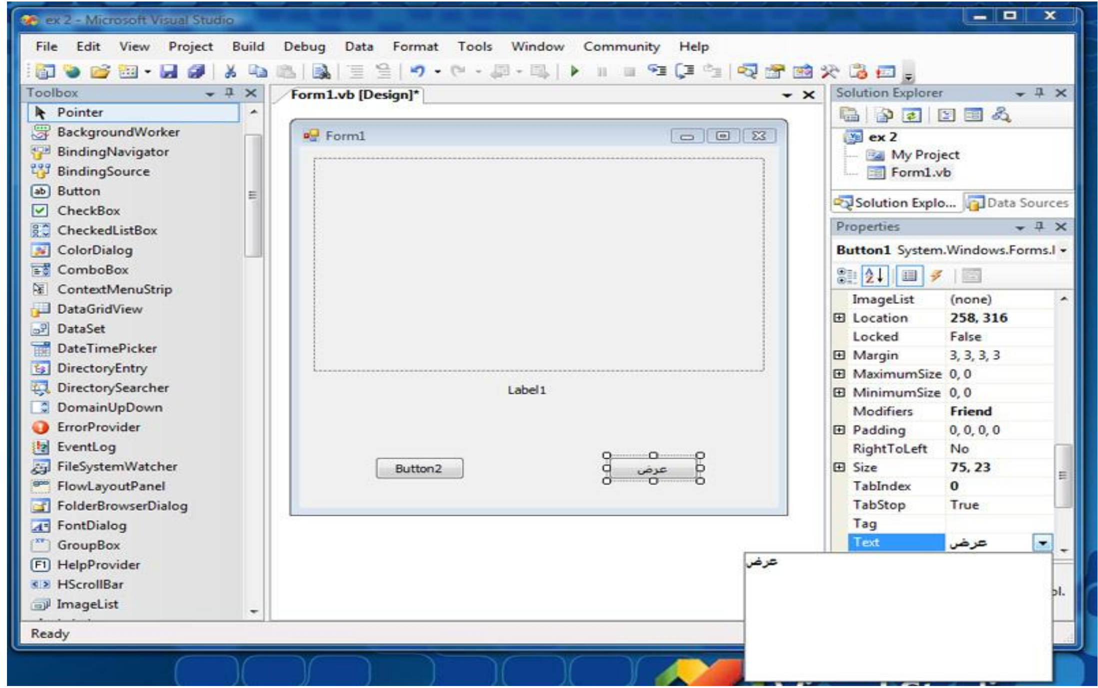


-قم بتغيير الخاصية SizeMode إلى StretchImage ومعنى ذلك أن الصورة ستظهر كاملة وستملئ الإطار

4- سوف نقوم الآن بتغيير بعض خصائص Label1

-اذهب إلى خصائص Label1 قم بتغيير Visible لتصبح False بدلا من True

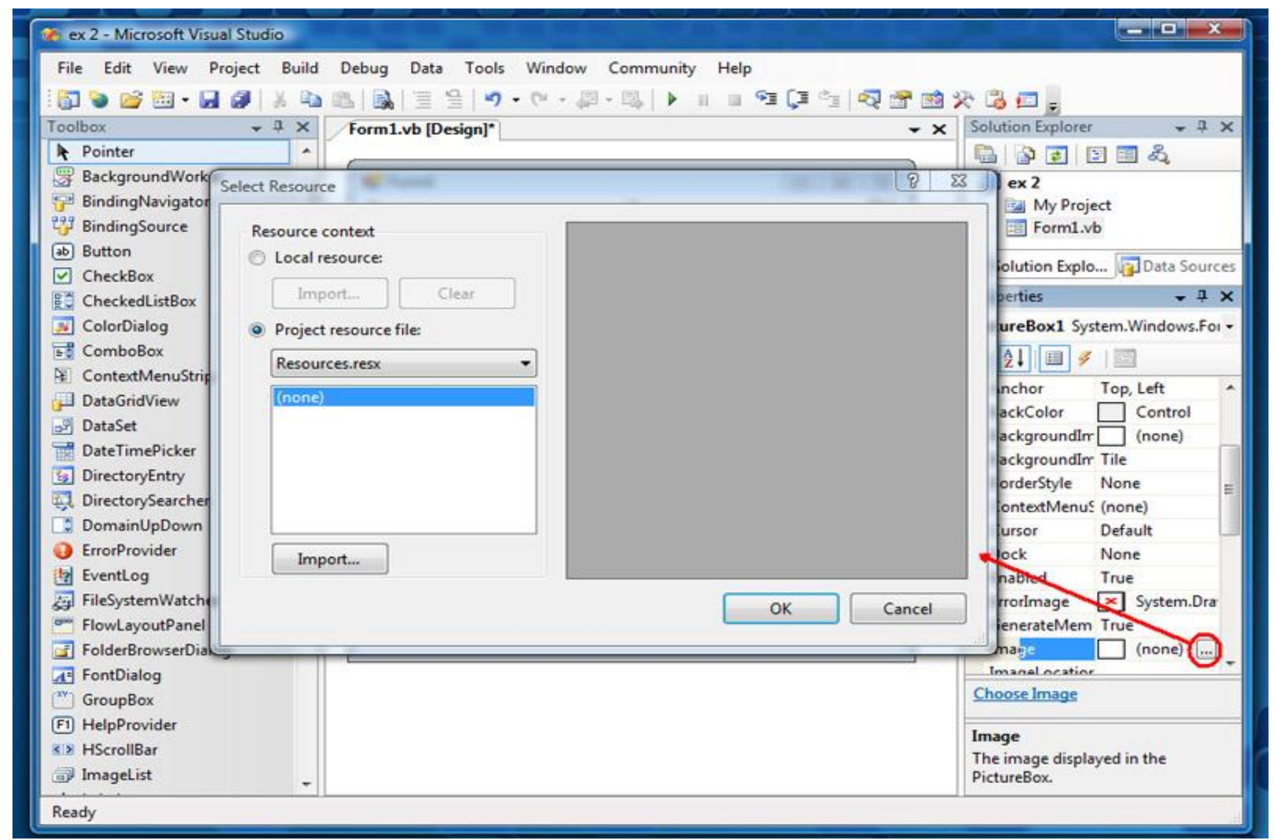
5- قم بتغيير اسم Button1 إلى عرض و Button2 إلى إنهاء وذلك عن طريق الذهاب إلى خصائص كل واحد منهما وتغيير الاسم من الخاصية Text ثم قم بتجربة البرنامج عن طريق F5



6- سوف نقوم الآن بإعطاء الوظائف لمكونات المشروع

-اذهب إلى خصائص PictureBox كما هو موضح بالنقطة رقم 3 ثم إلى Image ثم قم بالضغط على علامة المستعرض تظهر لك نافذة Select Resource اختيار المصدر قم بالضغط على Local Resource ثم Import وقم باختيار أي صورة من جهازك إيضاح

عند اختيارك الخيار الأخر وهو Project Resource File ثم Import سيتم إنشاء مجلد يسمى Resource يتم حفظ الصورة بداخلة ولن يتم دمج الصورة داخل الملف التنفيذي للمشروع



-اذهب إلى خصائص العنوان Label1 و في الخاصية Text قم بتغيير كلمة Label1 إلى اسم الصورة الذي اخترتها في الخطوة السابقة

7- كتابة الكود :

-ضغطة مزدوجة بالماوس على الزر عرض ثم قم بكتابة الكود التالي

كود:

```
PictureBox1.Visible = True
Label1.Visible = True
```

ومعنى هذا الكود

عند القيام بالضغط على هذا الزر قم بإظهار محتوى صندوق الصورة

عند القيام بالضغط على هذا الزر قم بإظهار محتوى العنوان Label1

-ضغطة مزدوجة بالماوس على الزر إنهاء وقم بكتابة الكود التالي

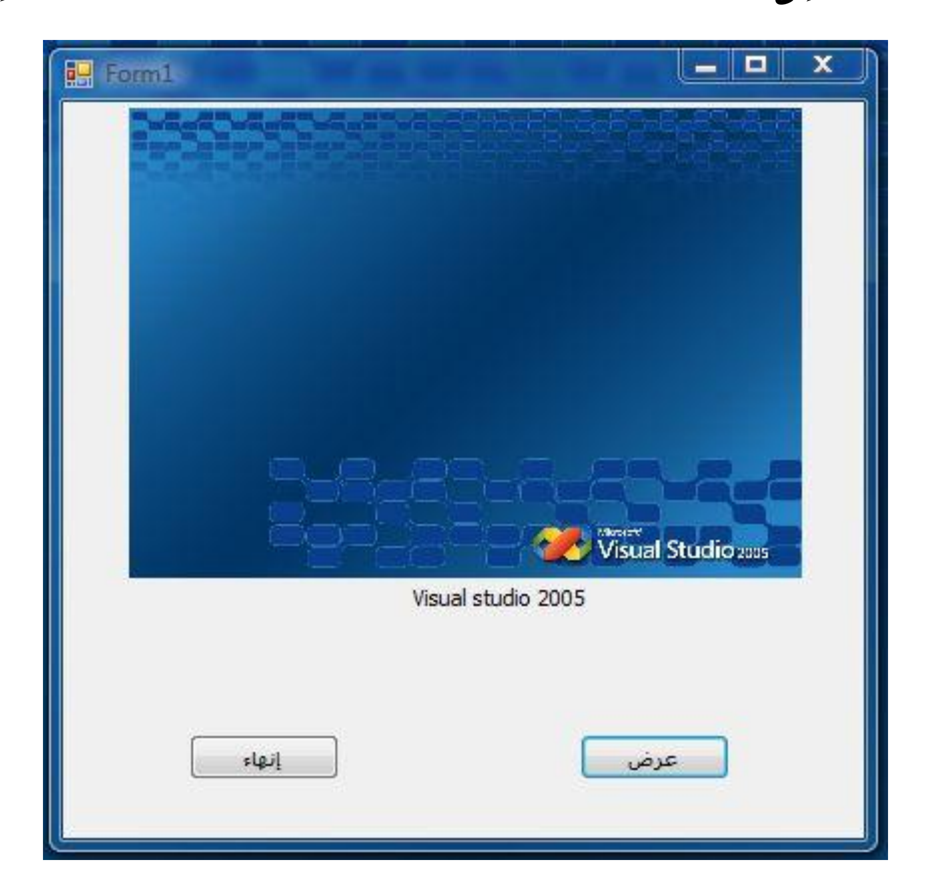
كود:

```
End
```

ومعناه واضح نهاية أو إغلاق

قم بتنفيذ المشروع بالضغط على F5

اضغط على عرض لرؤية الصورة وعنوانها و اضغط على إنهاء لإغلاق البرنامج

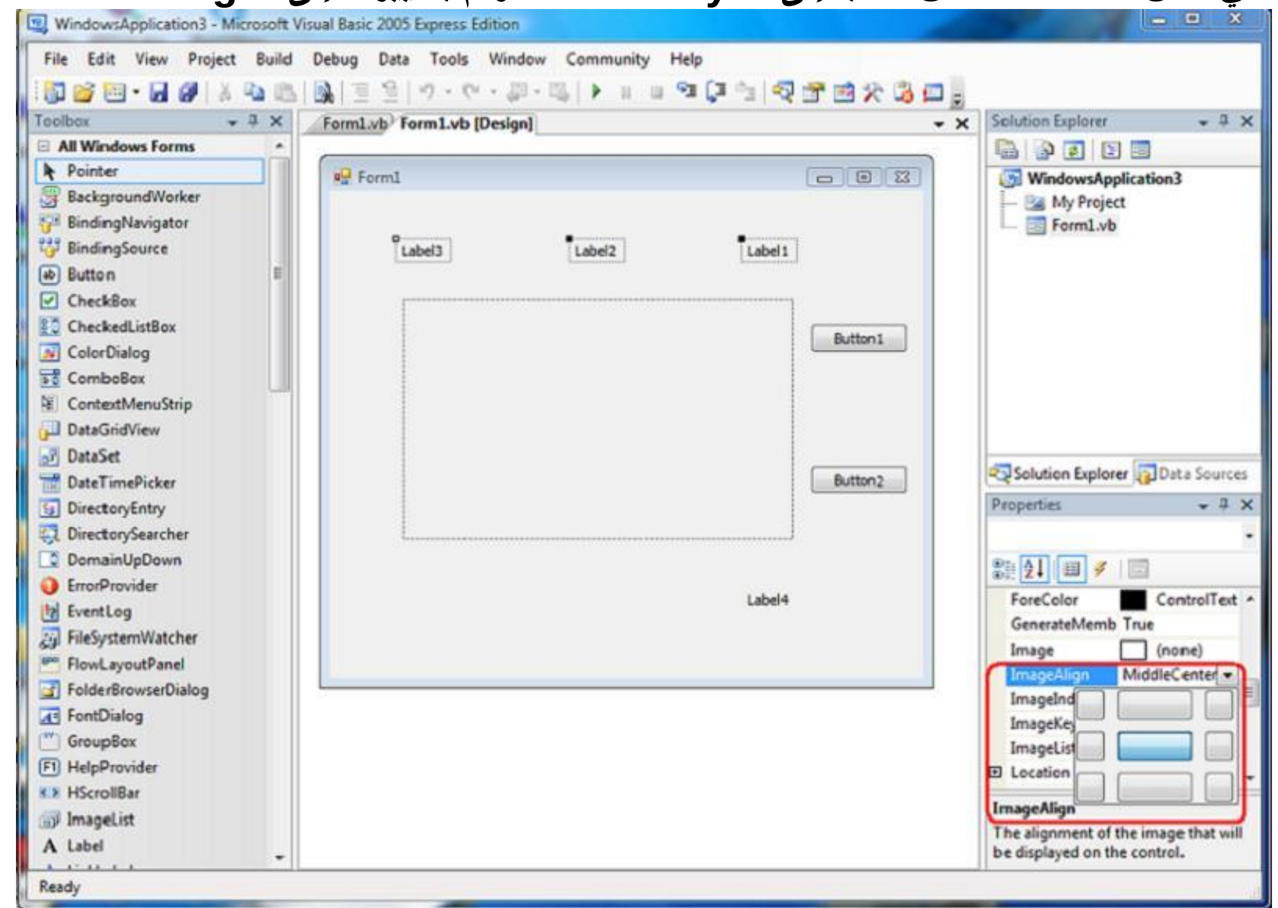


البرنامج الثالث : رقم الحظ

الهدف من المشروع
-توليد أرقام عشوائية
-إظهار الصورة المدمجة فقط عند ما يتم توليد رقم معين
سوف تظهر الصورة فقط إذا كان رقم 7 من ضمن الأرقام العشوائية الناتجة مع إعطاء صوت للتنبيه عند الحصول على الرقم الهدف
1- مشروع جديد
2- قم بإدراج عدد 2 Buttonو 4 Label و1 PictureBox يعنى2 زر و 4 عناوين ومربع صورة ونقوم بضبط الشكل العام
3- تخصيص المكونات
تغيير خصائص Label1 و Label2 و Label3
لكي لا تقوم بتخصيص كل عنوان على حدة و تخصيص العناوين الثلاثة دفعة واحدة اتبع ما يلي:
قم بعمل كليك على Lable1 ثم اضغط Shift ولا ترفع إصبعك من على Shift حتى تنتهي من التحديد
قم بعمل كليك على الزر الثاني ثم على الزر الثالث
الآن قم برفع إصبعك من على Shift . أي تخصيص نقوم بعملة الآن سينطبق على الثلاثة مجتمعين
-اذهب إلى نافذة الخصائص وقم بتغيير ImageAlign إلى Middle Center وهذا سيؤدى

إلى أن المحتوى سيظهر في المنتصف تماما

-في نفس نافذة الخصائص اذهب إلى BorderStyle وقم بتغييرها إلى FixedSingle



نقوم أيضا بتغيير خصائص الخطوط

-اذهب إلى الخاصية Font ثم اضغط على علامة + واسمها باللغة الإنجليزية ellipsis تفتح لك قائمة بخصائص الخط نقم بتغييرها كالتالي

قم بتغيير نوع الخط إلى Times New Roman أو كما تشاء

قم بتغيير Size حجم الخط إلى 25 أو كما تشاء

قم بتغيير نمط الخط Bold إلىTrue

-اذهب إلى الخاصية Text وقم بتغيير النص إلى 0 يعنى رقم صفر

-اذهب إلى الخاصية ForeColor وقم بتغيير اللون أو كما تريد

تغيير خصائص الأزرار

-قم بتسمية أحد الأزرار تشغيل والأخر إنهاء

تخصيص صندوق الصورة

اذهب إلى خصائص مربع الصورة ثم قم بتغيير الخصائص التالية

-اذهب إلى Image ثم قم بالضغط على علامة المستعرض تظهر لك نافذة Select Resourceاختيار المصدر قم بالضغط على Local Resource ثم Import وقم باختيار أي صورة من جهازك

-قم بتغيير الخاصية SizeMode إلىStretchImage

-قم بتغيير Visible لتصبح False بدلا منTrue

لمعرفة المزيد عن طريقة تخصيص مربع الصورة قم بالرجوع إلى البرنامج الثاني
تخصيصLabel4
-قم بتسمية العنوان باسم الصورة التي اخترتها وقم بتغيير لون الخط ونوعه.......الخ حسبما تريد
-قم بتغيير الخاصية Visible إلىFalse
4- كتابة الكود
-قم بالضغط ضغطة مزدوجة على أي منطقة خالية في الشكل Form1 واكتب الكود التالي

كود:

```
Randomize()
```

- ضغطة مزدوجة على الزر تشغيل وكتب الكود التالي
كود:

```
PictureBox1.Visible = False
Label1.Text = CStr(Int(Rnd() * 10))
Label2.Text = CStr(Int(Rnd() * 10))
Label3.Text = CStr(Int(Rnd() * 10))
If (Label1.Text = "7") Or (Label2.Text = "7") _
Or (Label3.Text = "7") Then
PictureBox1.Visible = True
Label4.Visible = True
Beep()
End If
```

-ضغطة مزدوجة على الزر إنهاء واكتب الكود التالي

كود:

```
end
```

يمكنكم وضع رقم أخر غير رقم7 للتجربة
قم بتطبيق العمل عن طريقF5



اضغط على الزر تشغيل حتى يتم ظهور أرقام تحتوى على الرقم 7 فتظهر الصورة وتسمع صوت التنبيه اضغط على الزر إنهاء ومعناها انتهى.

المثال الرابع الآلة الحاسبة البسيطة

-مشروع جديد

سنقوم الآن بالبدء بتخصيص الفورم قبل الشروع بالعمل والغرض من تخصيصه هو تغيير اتجاهه من اليمين إلى اليسار حتى يصبح الفورم بالصورة العربية

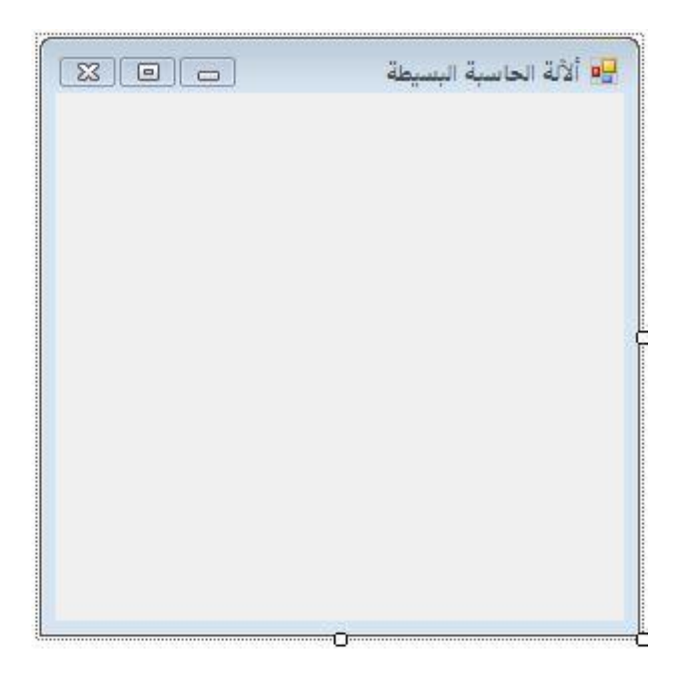
اذهب إلى خصائص الفورم Form1 وقم بتغيير الخصائص التالية

Reighte To Left من NO إلىYes

Reighte To Left layot منFalse إلىTrue

Textمن Form1 إلى ألآلة الحاسبة البسيطة

أصبح الفورم الآن قابلا للتعامل معه من اليمين لليسار بكافة مكوناته التي سنضعها بعد قليل



-من صندوق الأدوات نقوم بإدراج التالي

Textboxعدد 3

Labelعدد 3

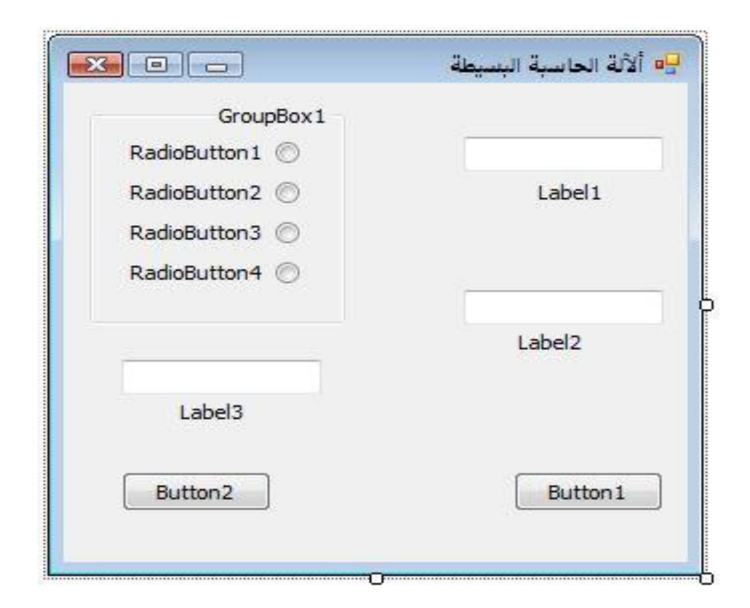
Buttonعدد 2

GroupBoxعدد1

Radio Buttonعدد 4

ثم نقوم بترتيبها بإدراج الأربعة Radio Buttons داخل مربع المجموعة Group Box وتنظيم باقي

المكونات بشكل مناسب وأرجوا منكم أن تقوموا بتنظيم العمل بشكل مناسب من ناحية الأشكال والخطوط والألوان.....الخ



نقوم بتسمية المكونات كالتالي

- ***GroupBox1عمليات***
- ***Radio Button1جمع(+)***
- ***Radio Button2طرح(-)***
- ***Radio Button3ضرب(*)***
- ***Radio Button4قسمة(/)***
- ***TextBox1الرقم الأول***
- ***TextBox2الرقم الثاني***
- ***TextBox3الناتج***



الآن الكود

-دبل كليك على الفورم ثم في أسفل السطر التالي Public Class Form1

كود :

```
Dim FirstNum, SecondNum As Double
```

-دبل كليك على الزر تنفيذ ثم

كود :

```
FirstNum = TextBox1.Text

SecondNum = TextBox2.Text
```

مطلوب الآن أن نقوم بوضع أكواد 4 عمليات بعد الكود السابق مباشرة وهى الأكواد التـي تصف عمليات الجمع والطرح والقسمة والضرب وسأقوم أنا بوضع كود لعملية واحدة وعليكم توقع الأكواد الثلاثة الباقية

كود عملية الجمع

كود :

```
If RadioButton1.Checked = True Then

TextBox3.Text = FirstNum + SecondNum

End If
```

كود عملية الطرح

كود :

```
If RadioButton2.Checked = True Then
TextBox3.Text = FirstNum - SecondNum
End If
```

كود عملية الضرب

كود:

```
If RadioButton3.Checked = True Then
TextBox3.Text = FirstNum * SecondNum
End If
```

كود عملية القسمة

كود:

```
If RadioButton4.Checked = True Then
TextBox3.Text = FirstNum / SecondNum
End If
```



لا تنسوا الزر End الذي يعنى انتهى